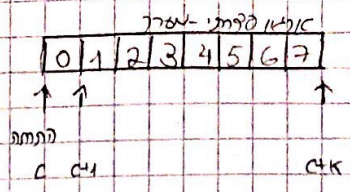


סיפורי זמן ותוכן ואי-אזניות

⇒ (הקצאה צבאית)



⇒ הקצאה סטטי



* הקצאה סטטי-משהו שא משתנה, מקצים לאזו קינול

* מחלוקת של העיקר שהוא אזו צבאית

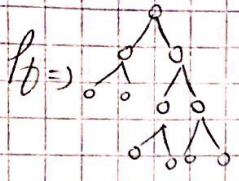
* זהו חתירה של ציפון יש כמות

* כל זה פיסוק מורכב מקטעים והפניסיו מורכבים מסיקור

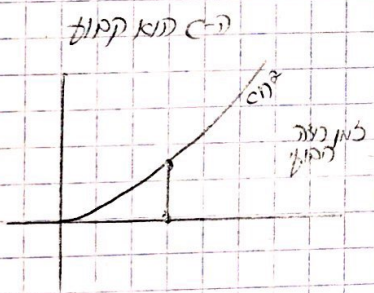
* היתרון של העיקר שהוא מאזן יעיל

* בחירת מקנה הומונים משנה על יעילות ההתקופ

* יעילות אחרות = זמן לינה (למון ריבוי משנה אזי עם הנוענים).



היסט ביו-זמן זמן אזור קטט הוא n^2 (מפני אזוריות עם סיקור)



קט: 2 נענים
למון ריבוי: 4 שניות

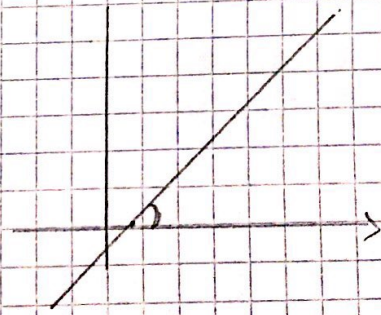
קט: 2 נענים
למון ריבוי: $2^2 = 4$ שניות

קט: 4 נענים
למון ריבוי: $4^2 = 16$ שניות

קט: 5 נענים
למון ריבוי: 25 שניות

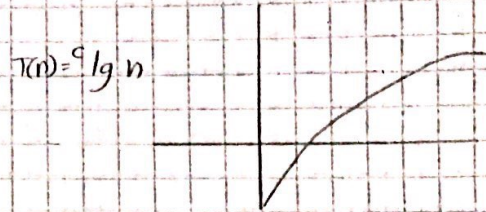
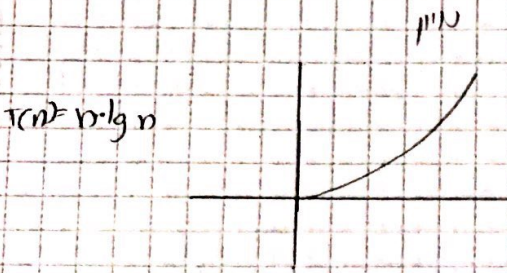
סיבוכיות זמן
אזויות A
זמן לינה T(n)
כאשר n הוא אזור הקטט
סיבוכיות האי-אזניות A
נענה אזי העיקר הנוענים T(n)
-אזויות:

$T(n) = c \cdot n + d$
קטלים: c, d
 $n = 1, 2, 3, \dots$





למען כרוב זריזות



סיבוכיות מקומית (אנחנו בקובץ הזה)
כמות הזיכרון הנדרש (הזרם) היא אוליגוריתמית

לדוגמה
מס: זיכרון עבור נתונים
מ.ד: זיכרון נוסף

סימון אסימפטוטי:

(בגודל)	}	מספר קטן \rightarrow	O קטנה	O קטנה
		מספר מתמון \rightarrow	ω קטנה	ω קטנה
		מספר 13-13 \rightarrow		ω קטנה

$T(n) = 2n \lg n + 4$

$T(n) = (n+1) \cdot \lg 4$
 $T(n) = O(n \lg n)$

$T(n) = n+3$

$T(n) = 2n + \lg - 2$

$T(n) = \lg n + n$

$T(n) = \lg n + 3$

$T(n) = O(n)$

$T(n) = \lg(m^2)$

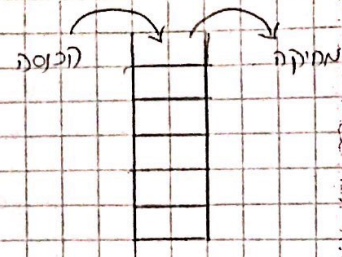
$T(n) = \lg n + \lg \lg n$

$T(n) = 3 \lg n - 2$

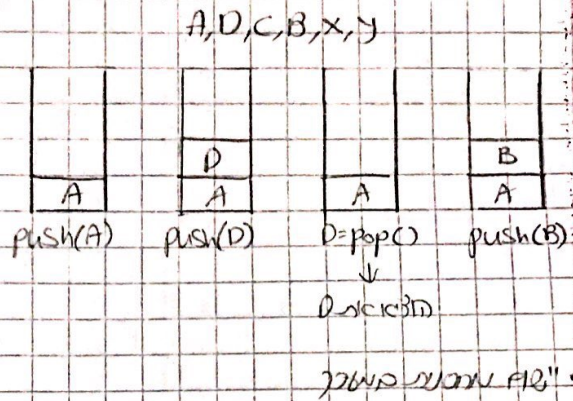
$T(n) = O(\lg n)$

STACK = מיון

LIFO - ניתן להוסיף רק את האחרון שנכנס
Last in, first out



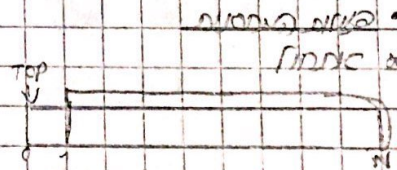
כמה פעמים הוספה ופחות פעמים הסרה
push - הוספה
pop - הסרה



push ארץ ישראל - [I s r a e l] [H a y o l i] *

(overflow) - עברנו את גודל המיון = אירוע

מחסור מקום = מיון קטן ממימד, זה קרה מימד (underflow)



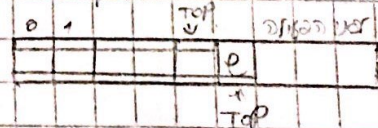
מחסור מקום - TOP(S) = 0 -> A[-1+1]

אם יש מקום להוסיף אם TOP(S) = N ->

הוספת איבר
push(S, e)

TOP(S) = TOP(S) + 1

TOP(S) = e



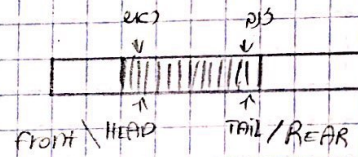
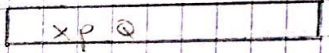
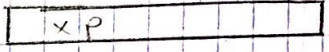
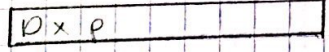
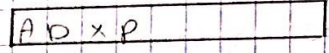
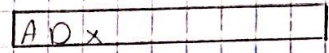
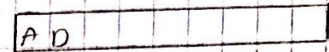
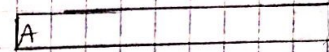
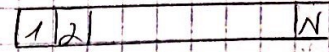
26/10/18

* מחקה (ראיב) בנס אורווו $pop(S)$

if $Top(S) = 0$ מחקה
 $x = e$ קוצא
 $Top(S) = Top(S) - 1$

מחקה נונד אור: $QUEUE =$ מנג
FIFO קעור

ACNS "מנס סמנג"



: מנס

ENQUEUE : מנס x

$TAIL = TAIL + 1$

$Q[TAIL] = x$

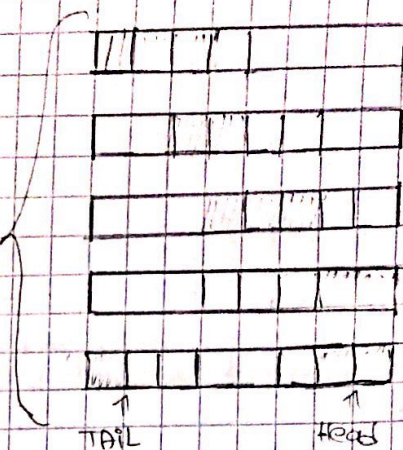
DEQUEUE : מחקה x

$y = Q[HEAD]$

$HEAD = HEAD + 1$

20/10/18

1 קבוצת מידע - רשימה



2/11/18

- * תמיכה - חוקה ישנו כאשר מנסים לתחוק איתה מחסנית ביקה
- * גישה - גישה שלן כאשר מנסים להיגים איה עם מחסנית מאנה
- * במקרה של יישום במזין: מצב גישה הוא מאק מאן
- * כמות בזיקה:
- האם המחסנית חוקה?
- האם המחסנית מאנה?

- * מחסנית ביקה - מחסנית חוקה כאשר $Top = 0$
- * עם מקומות של איברים שניתן להכניס למחסנית הוא Max
- * $Top = 0$ - האם האברים הנמצאים במחסנית בסוף

push (S, x) *

בתחילתה זרימה של המחסנית

if $top = Max$ error overflow
else =

$top = top + 1$

$A[top] = x$

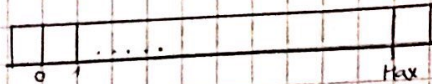
pop (S) *

if $top = 0$ error underflow
else

$x = A[top]$

$top = top - 1$

לביצוע קצור מנר עם יחסים במנר
 עם המחסנית: מנר = (QUEUE) מנר (או A)
 לבניית קבוצה אחת, מחיקה מניצב המנר



הכנסות - המנוחה בה לא ניתן להכניס יותר הנכנסות כי אינן נכנסות בסדרה
 ניתן להכניס רק מחיקות.



אובדן יציב של יציב (מחיקות הליניאר) של מנר במנר (Circular array)
 מחיקות במנר (ליו) $A[1+Max]$ כאשר חיסור האנדקס מספיק קטנה במנר פשוט (Modulo)
 (Mod)

$$1 \leq i < Max \Rightarrow i+1 \leq Max$$

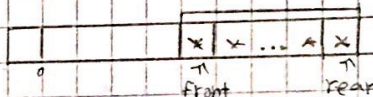
$$i = Max \Rightarrow i+1 > Max$$

$$i < Max \Rightarrow i \text{ Mod } Max + 1 = i + 1$$

$$i = Max \Rightarrow i \text{ Mod } Max + 1 = 1$$

QUEUE: fifo

Front, head : כאשר המנר
 rear, tail : במחיקות

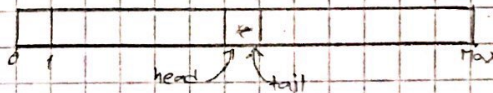


מחיקות כאשר המנר הוא קבוצה של מנר (מחיקות במנר) (מחיקות במנר)
 מחיקות במנר כאשר המנר הוא קבוצה של מנר

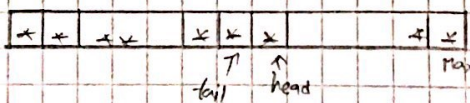
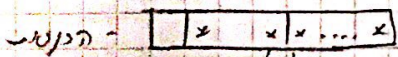
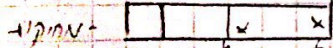
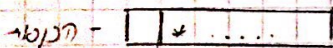
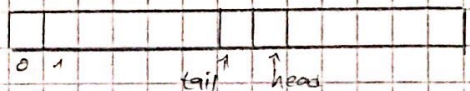
מחיקות במנר כאשר המנר הוא קבוצה של מנר



מחיקות במנר כאשר המנר הוא קבוצה של מנר (Mod Max)

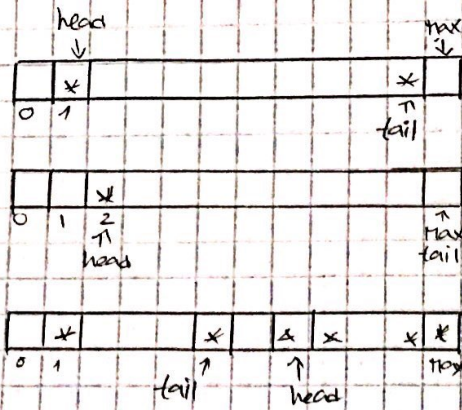


מחיקות במנר כאשר המנר הוא קבוצה של מנר head=tail מנר כיון.



מחיקות במנר כאשר המנר הוא קבוצה של מנר
 מחיקות במנר כאשר המנר הוא קבוצה של מנר
 מחיקות במנר כאשר המנר הוא קבוצה של מנר

מחיקות במנר כאשר המנר הוא קבוצה של מנר



קודם (מספרות) אולי נעשה קונסטרוקטור

head = 1; tail = Max - 1 (1)

head = 2; tail = Max (2)

tail = head - 2 (הי (1)) (3)

יבוא על מור כמספר מלאי
 Count = קבוצת מאונן (1)

- Q: קבוצת מור
- A[1+max]: גודל
- head: אינדקס של ראש
- tail: אינדקס של זנב
- count: מספר איברים
- count = 0: מור ריק
- count = max - 1: מור מלא

Max: יבוא קבוצת מור קבוצת 1 ו-Max
 Max: יבוא קבוצת מור קבוצת 0 ו-Max
 Initialize (Q) מתחילת

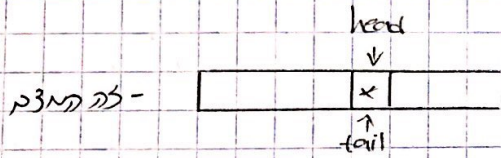
```

head = 1
tail = 0
count = 0
: AddQueue (Q, x) הוספת איבר
if count == max
    Error Overflow
else
    count = count + 1
    tail = (tail Mod Max) + 1
    A[tail] = x
: DeleteQueue (Q) מחיקת איבר
if count == 0
    Error Underflow
else
    count = count - 1
    x = A[head]
    head = (head Mod Max) + 1
    
```


Queue operations

```

Q: array
A[1+max] : array
head : int
tail : int
: Initialize (Q)
head = 1
tail = 0
: AddQueue (Q, x)
if ((tail = head - 1) and (tail > 0)) or
(tail = max) and (head = 1))
Error overflow
else
tail = (tail Mod max) + 1
A[tail] = x
: DeleteQueue (Q)
if tail = 0
Error underflow
else
x = A[head]
if head = tail
head = 1
tail = 0
else
head = (head Mod max) + 1
    
```



2/11/18

מבנה נתונים (קבוצה) - רשימה (קבוצה) - רשימה (קבוצה) - רשימה (קבוצה)

DOUBLE Queue = DEQUE (רשימה כפולה)

$head = head - 1$ (עבר) = Add Head (Q, x)
 $tail = tail + 1$ = Add Tail (Q, x)
 $head = head + 1$ = Delete Head (Q)
 $tail = tail - 1$ (עבר) = Delete Tail (Q)

- קבוצה
- * קבוצה קבוצה
- * מחיקה מחיקה
- * קבוצה קבוצה
- * מחיקה מחיקה

רשימה כפולה



2/11/18

אורי פנקס המורה
pincasur@gmail.com
שאלה קודם - שאלה תיאורית

מחסנית

1) קבוצה של מספרים המכילה מספרים שונים, האם ניתן לבנות אותה באמצעות תוספת והסרה?

STACK

```

push(S,3) 10
push(S,-4) 9
print(pop(S)) 8
push(S,2) 7
push(S,-1) 6
print(pop(S)) 5

```

המספרים הם אלו שהיו בתחילת הריבוע
קצתם רק קטנים מהמספרים האחרים.

```

while (not (Empty(S))) do
  print(pop(S))

```

(11/16) אורי פנקס

מספר	מחסנית	הצורה
1	[3]	-
2	[3, -4]	-
3	[3]	-4
4	[3, 2]	-4
5	[3, 2, -1]	-4
6	[3, 2]	-4, -1
8 ₁	[3]	-4, -1, 2
8 ₂	[]	-4, -1, 2, 3

2) קראים מהקלט נשמה של n מספרים. מוציא ויוצא כי $\frac{n}{2}$ מספרים הם זוגיים $1 - \frac{n}{2}$ או זוגיים.
 צנוש ארבעים את n המספרים, קוצים אתם הזוגיים ואחריו אתם הזוגיים.
 (אם חשבונית/סדר קיין הזוגיים או קיין הזוגיים).

שמה השתחשטות:

S₁: מאחסן המספרים הזוגיים

S₂: מאחסן המספרים הזוגיים

אחר אתם המספרים ענף ארבעים בסדר החוקים

Ex

$O(n)$ - מספר הזוגיים

$O(n)$ - מספר הזוגיים

$O(n)$

1. for $i \leftarrow 1$ to n do
2. $x \leftarrow \text{READ}$
3. if $(x \bmod 2 = 0)$ then
4. push(S_1, x)
5. else
6. push(S_2, x)
7. for $j \leftarrow 1$ to $\frac{n}{2}$ do
8. print(pop(S_1))
9. for $k \leftarrow 1$ to $\frac{n}{2}$ do
10. print(pop(S_2)).

סיבוכות כוללת $O(n)$

3) קראים מהקלט נשמה של n זוגיים. פרוש ארבעים את החלוקה בסדר הפוך

Ex3

1. for $i \leftarrow 1$ to n do
2. $x \leftarrow \text{READ}$
3. push(S, x)
4. for $j \leftarrow 1$ to n do
5. print(pop(S))

סיבוכות כוללת $O(n)$

4) שונה נשימה של n קטגוריות
 תאורת מחסנית היא סדרה מוגבלת המתקבלת מהרשימה $1, 2, \dots, n$.
 כאשר כל מספר מופיע אחת מלבד פעם אחת.
 * רצפתה של המספר הנכנס בהרשימה
 * הכנסת המספר הנכנס בהרשימה למחסנית
 * הוצאת איקו מהמחסנית והרצפתו. אם המחסנית ריקה.

א. $n=1$

תמונה אפשרית: 1 (כיש בהרשימה רק 1)

ב. $n=2$

תמונה אפשריות:
 1) רצפתה 1, רצפתה 2 : 1, 2
 2) רצפתה 1, רצפתה 2, הוצאה מהמחסנית והרצפתה 1 : 2, 1

אז אפשרי

(3, 1, 2)

ג. $n=3$

תמונה אפשריות:
 1) רצפתה 1, רצפתה 2, רצפתה 3 : 1, 2, 3
 2) רצפתה 1, רצפתה 2, רצפתה 3, הוצאה מהמחסנית והרצפתה 1 : 2, 3, 1
 3) רצפתה 1, רצפתה 2, רצפתה 3, הוצאה רצפתה 1, הוצאה מהמחסנית : 1, 3, 2
 4) רצפתה 1, רצפתה 2, רצפתה 3, הוצאה מהמחסנית, הוצאה מהמחסנית, הוצאה מהמחסנית : 3, 2, 1
 5) רצפתה 1, רצפתה 2, הוצאה מהמחסנית והרצפתה 1, רצפתה 3 : 2, 1, 3

5) קובץ המקלט (השייח) n סימני סוגרים ")", "(", צריך לקבל (אם השייח מתאמת) זמון-סוגרים.
 return

return $(a+b)(c+d) + (e+f)$ חוקי. כל סוגרים שכתבנו, סגורו.

Ex 4

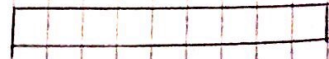
```

1. for i ← 1 to n do
2. x ← READ
3. if (x = "(") then
4. push(S, x)
5. else
6. if (Empty(S)) then
7. return "ERROR"
8. else
9. pop(S)
10. if (Empty(S)) then
11. return "RIGHT / True"
12. else
13. return "False / wrong"

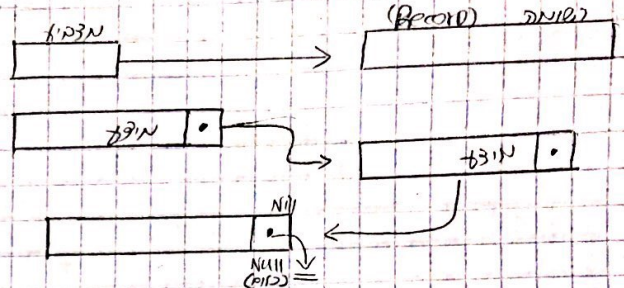
```


- Linked Lists

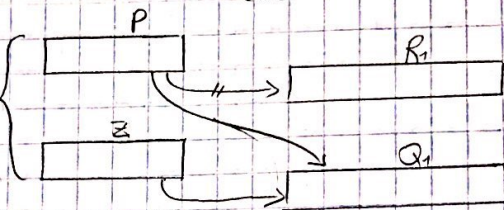
מתי: ליסטת



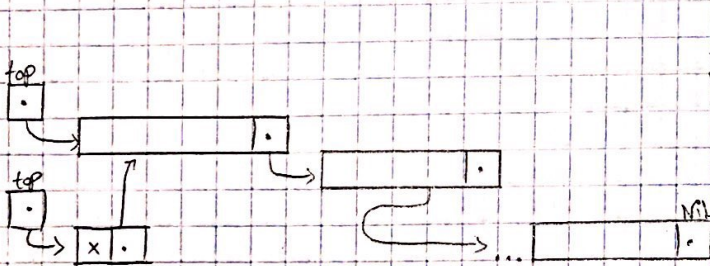
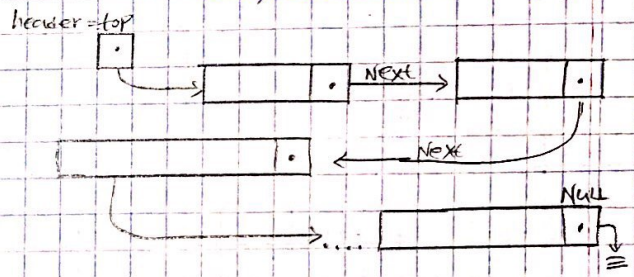
אי אפשר להציגו מלבד, אלא יצור מלבד גבוה ונחלקו את הנתונים מהיכן ואיפה (רצף-יחיד).
 מסודרות: כמו רשימה, בעמוד שני היכול.
 - ביכול (הקצאת) ביחידות - או רצף
 • ליסטת או רצף - מסודר
 • הקצאת ביחידות - יחידות
 מתי: (Pointer) הוא משהו שקובץ הוא כתיבת



$P \neq Q$ אם הם שונים אך הם יכולים להצביע על אותה רשימה
 ואם הריבוי של R_1 (אם)



* מחברת: יישום הרשימה מחוברת



מתי: אחרת - Top ומחברת את הרשימה החברה או קצף הרשימה
 והאישור להיות הרשימה השנייה (P = Top מחברת)

push(S: x) - הוסיף רשומה בתוכנית את x, מתחילת אוסף רשומות.

```

push(S: x)
new(p)
p->info = x
p->next = null
pushNode(S: p)
↓
הוסיף רשומה חדשה

```

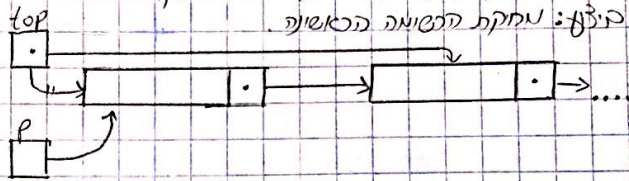
```

pushNode(S: p)
=> if p = null
    Error
    if top = null
        top = p
    else
        p->next = top
        top = p

```

↓
לרבות רשומות החדשה זכרון
(המחשבת)

- מחקה מחקת על צומת (רשומה) השנייה מקובלת
היציג: מחקת הרשומה הבאשונה.



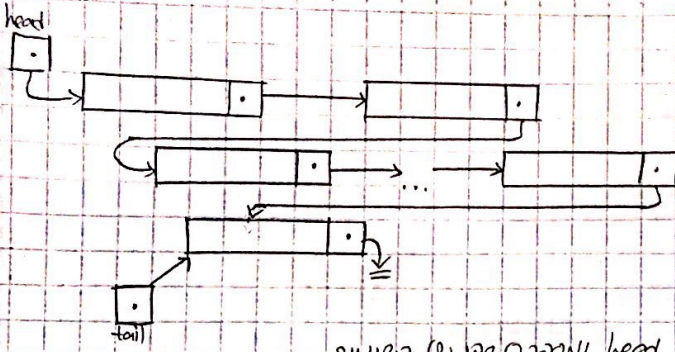
מתקיים את ה top מהרשומה ומסמך הנתון היש
וכי Top מצביע לרשומה הבאה (הבאשונה) (מחקת)
Top = Top->Next

```

popNode(S)
if top = null
    Error
else
    p = top
    top = top->next

```

pushNode(S: p) ! popNode(S) כשני המחקים קיימים סיכוי של איבדנות. שתי הפעולות לבנות סימן (אם)
דומה, שני (קודם) (סימנים) (מחקת)

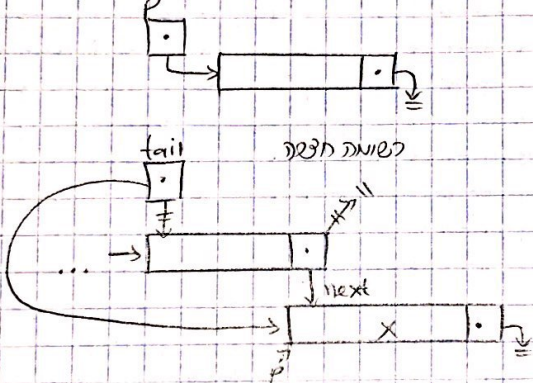


* ערך: יישום רשת סגורה מקובלת.
 רשת מקובלת - מחיקה קבועה
 רשת סגורה

מקרה פרטי של רשת סגורה של רשתות head, tail ומספר קשרים של רשתות

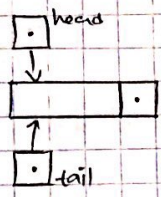
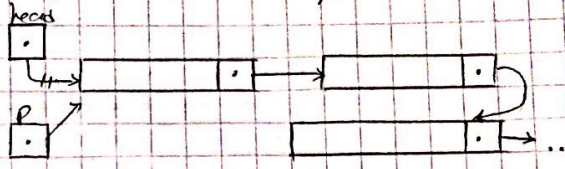
אתחול רשת:
 InitializeQueue(Q)
 Q → head = null
 Q → tail = null
 O(1)

הוספה: הוספת רשת (רשת) רשת (רשת) Tail



הוספת רשת:
 AddNode(Q, p)
 if p = null
 Error
 else
 if head = null
 head = p
 tail = p
 else
 tail → next = p
 tail = p
 O(1)

מחיקת מחקר, מחיקת (מחיקת) את הכמות בסיסית.



```

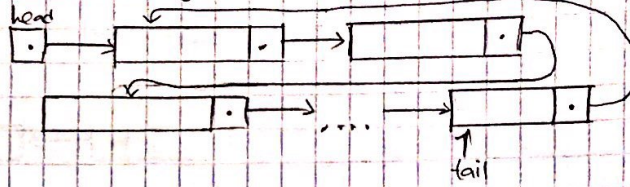
Delete Node (x)
if head = null
Error
else
p = head
head = head->next
if head = null
tail = null

```

Simply-Linked List

* מחיקת מחקר היא פקודת מחיקת או פקודת מחיקת מחקר

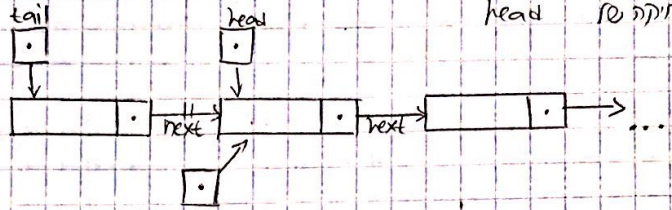
Circular Linked List רשימת מקושרת מעגלית



```

tail->next = p
p->next = head
tail = p
O(n)
    
```

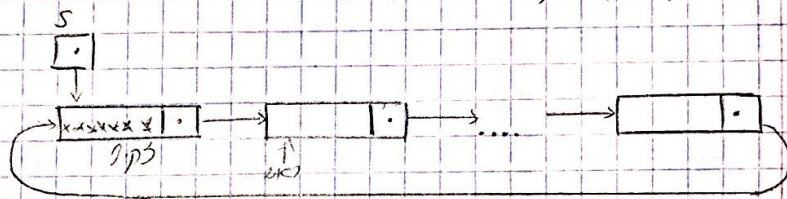
head מחזיקה רש



```

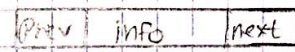
head = head->next
O(1)
    
```

Sentinel רשימת מקושרת עם צ'ק'ן

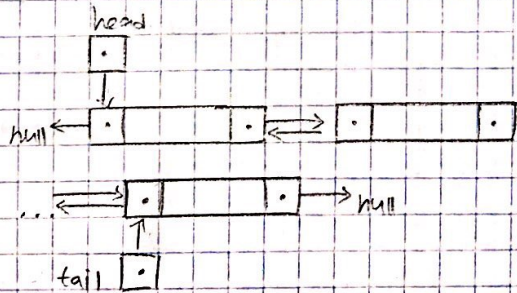


הכוננים של הצ'ק'ן חסבי משמעות הצ'ק'ן נעזר בצ'ק'ן מעביר מידע (למשל רשימה ריקה) או צורך בסיסי מידע של רשימה ריקה.

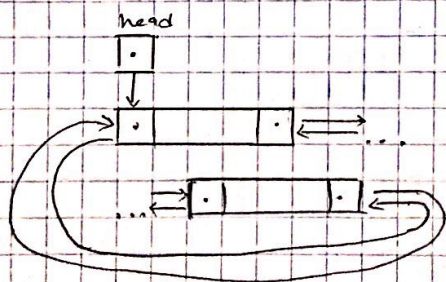
Doubly Linked List רשימת צו-מקושרת



next : מצביע אל הצ'ק'ן
prev : מצביע אל הקודם



רשימת צו-מקושרת מעגלית



הערה: המערכת היא בעלת זיכרון מוגבל. כל פעולה על המערכת תבוצע רק אם יש לה זיכרון מספיק.

```

push (S, 7) - א
push (S, 4) - ב
enqueue (Q, 1) - ג
print (pop(S)) - ד
enqueue (Q, pop(S)) - ה
push (S, dequeue(Q)) + dequeue(Q) - ו
push (S, 2) - ז
enqueue (Q, 8) - ח
push (S, pop(S) * dequeue(Q)) - ט
while (not (empty(S))) do - י } 3np
    print (pop(S)) - יא
while (not (empty(Q))) do - יב } 3np
    print (dequeue(Q)) - יג
    
```

הערה: המערכת היא בעלת זיכרון מוגבל.

הצגה	מ	הצגה	הזמן								
-	<table border="1"><tr><td></td><td></td><td></td></tr></table>				<table border="1"><tr><td>7</td><td></td></tr></table>	7		1			
7											
-	<table border="1"><tr><td></td><td></td><td></td></tr></table>				<table border="1"><tr><td>7</td><td>4</td></tr></table>	7	4	2			
7	4										
-	<table border="1"><tr><td></td><td>1</td><td></td></tr></table>		1		<table border="1"><tr><td>7</td><td>4</td></tr></table>	7	4	3			
	1										
7	4										
4	<table border="1"><tr><td></td><td>→</td><td></td></tr><tr><td></td><td>1</td><td></td></tr></table>		→			1		<table border="1"><tr><td>7</td><td></td></tr></table>	7		4
	→										
	1										
7											
4	<table border="1"><tr><td></td><td>7</td><td>1</td></tr></table>		7	1	<table border="1"><tr><td></td><td></td><td></td></tr></table>				5		
	7	1									
4	<table border="1"><tr><td></td><td></td><td></td></tr></table>				<table border="1"><tr><td>8</td><td></td></tr></table>	8		6			
8											
4	<table border="1"><tr><td></td><td></td><td></td></tr></table>				<table border="1"><tr><td>8</td><td>2</td></tr></table>	8	2	7			
8	2										
4	<table border="1"><tr><td></td><td>8</td><td></td></tr></table>		8		<table border="1"><tr><td>8</td><td>2</td></tr></table>	8	2	8			
	8										
8	2										
4	<table border="1"><tr><td></td><td></td><td></td></tr></table>				<table border="1"><tr><td>8</td><td>16</td></tr></table>	8	16	9			
8	16										
4, 16	<table border="1"><tr><td></td><td></td><td></td></tr></table>				<table border="1"><tr><td>8</td><td></td></tr></table>	8		10-11			
8											
4, 16, 8	<table border="1"><tr><td></td><td></td><td></td></tr></table>				<table border="1"><tr><td></td><td></td></tr></table>			10-11			
4, 16, 8	<table border="1"><tr><td></td><td></td><td></td></tr></table>				<table border="1"><tr><td></td><td></td></tr></table>			12-13			

Q קבלת מיושם של מני סוגים שתי מחסנות
 (שני סוגי מחסנות):
 S1 - מחסון איברי קטן
 S2 - מחסון איברי גדול

$O(1)$ { Empty(Q):
 1. return (EMPTY(Q.S1))

$O(1)$ { Full(Q):
 1. return (Full(Q.S1))

$O(1)$ { ENQUEUE(Q,X)
 1. push(Q.S1,X)

DEQUEUE(Q)
 1. while (not(EMPTY(Q.S1))) do
 2. push(Q.S1, pop(Q.S1))
 3. x ← pop(Q.S2)
 4. while (not(EMPTY(Q.S2))) do
 5. push(Q.S1, (pop(Q.S2)))
 6. return x

לוקח ה איברי קטן Q

(B) קבלת מיושם של מחסנות סוגים שני טורים
 (שני סוגי טורים):
 Q1 - מחסון איברי קטן
 Q2 - מחסון איברי גדול

$O(1)$ { Empty(S)
 1. return (EMPTY(S.Q1))

$O(1)$ { Full(S)
 1. return (Full(S.Q1))

$O(1)$ { PUSH(S,X)
 1. ENQUEUE(S.Q1,X)
POP(S)

$O(1)$ {
 1. x ← DEQUEUE(S.Q1)
 2. while (not(EMPTY(S.Q1))) do
 3. ENQUEUE(S.Q2,x)
 4. x ← DEQUEUE(S.Q1)
 5. while (not(EMPTY(S.Q2))) do
 6. ENQUEUE(S.Q1, DEQUEUE(S.Q2))
 7. return x

קבלת מיושם של מחסנות סוגים שני טורים
 (שני סוגי טורים):
 Q1 - מחסון איברי קטן
 Q2 - מחסון איברי גדול

לוקח ה איברי קטן Q

13) A 77077 ENINU 717-13 7077

LENQUE (DQ, x)

1. if $li = 1$ then
2. $li \leftarrow n$
3. else
4. $li \leftarrow li - 1$
5. $A[li] \leftarrow x$

RENQUE (DQ, x)

1. if $ri = n$ then
2. $ri \leftarrow 1$
3. else
4. $ri \leftarrow ri + 1$
5. $A[ri] \leftarrow x$

LDEQUE (DQ)

1. $x \leftarrow A[li]$
2. if $li = n$ then
3. $li \leftarrow 1$
4. else
5. $li \leftarrow li + 1$
6. return x

RDEQUE (DQ)

1. $x \leftarrow A[ri]$
2. if $(ri = 1)$ then
3. $ri \leftarrow n$
4. else
5. $ri \leftarrow ri - 1$
6. return x

חיפוש בתוך תבנית נתונה
חיפוש בתוך תבנית נתונה

חיפוש בתוך מערך

* חיפוש אינארי



מקרה טוב: יציאת אחריותם במקום הראשון
מקרה בינוני: מציאת אסרן ו הפיקוד
מקרה גרוע: יציאת במקום האחרון (סדר)

חיפוש כושל: המשתד לא נמצא באר מקום הסדרה (סדר 2 של מקרה גרוע)

- אם מציאת את המשתד במקום ה- i , $1 \leq i \leq N$, חיפוש ארץ i השואות ואז זמן כליה $O(1)$
- אם מציאת את המשתד במקום הראשון $i=1$, מקרה טוב, אז זמן כליה $O(1)$ - קטנה
- אם מציאת אסרן $i=N$, מקרה גרוע, אז זמן כליה $O(N)$ - אינארי
- בחיפוש כושל זמן כליה תמיד $O(N)$

השוואות:

$$\begin{matrix} 1+N = N+1 \\ 2+(N-1) = N+1 \\ \vdots \\ \vdots = N+1 \end{matrix} \left. \vphantom{\begin{matrix} 1+N = N+1 \\ 2+(N-1) = N+1 \\ \vdots \\ \vdots = N+1 \end{matrix}} \right\} \frac{N}{2}$$

$$1+2+\dots+N = \frac{N \cdot (N+1)}{2}$$

$$\frac{N}{2} \cdot (N+1)$$

$$\frac{N \cdot (N+1)}{2} = \frac{N+1}{2} \cdot N$$

השוואה 1	1
	N
השוואה 2	1
	N
i השוואות	1
	N
N השוואות	1
	N

זמן המהלך של השוואות בחיפוש של משתד נתון בסדר

$$N = O(N), \quad \frac{N+1}{2} = O(N), \quad 1 = O(1)$$

במציאת המינימום

LINEAR SEARCH = חיפוש ליניארי

Sequential Search = חיפוש סדרתי

L: סדרת מספרים (מסומם)
"L" = מספר

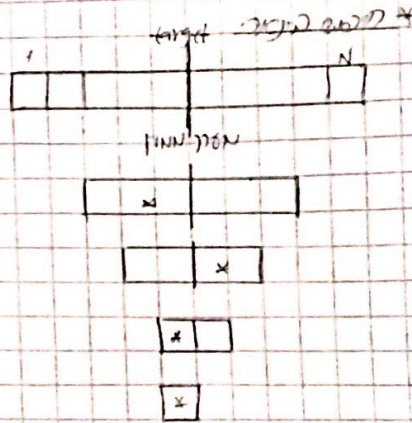
מטרה: Target
L [1...count] : מספר המספרים
(True/False) : found
1 ≤ i ≤ N : i

Sequential Search

```

O(n) {
  found = false
  i = 1
  while (not found and i <= count)
    if (L[i].key = Target)
      found = True
    else
      i = i + 1
}

```

דוגמה נוספת של חיפוש בינארי

מספר יונים $\approx 2^0$
 מספר יונים $\approx 2^1$
 מספר יונים $\frac{N}{2}$
 מספר יונים $\frac{N}{4}$
 מספר יונים $\frac{N}{8}$

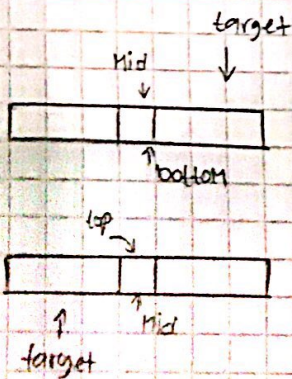
מספר יונים $\frac{N}{2^k}$...

$$N \approx 2^k$$

$$N \approx 2^k$$

$$O(\log n) \text{ מספר יונים} = (k - \text{מספר היונים}) \log_2 N = k$$

מספר יונים: 1
 מספר יונים: 2
 מספר יונים: 4
 מספר יונים: 8
 מספר יונים: 16
 מספר יונים: 32



Binary Search 1

```

bottom = 1
top = n
while (top > bottom)
    mid = (bottom + top) / 2
    if (target > L[mid].key)
        bottom = mid + 1
    else
        top = mid
if (top == 0)
    found = false
else
    if (L[top].key == target)
        found = true
    
```

else found = false
 location = top
 true km found km-
 km תמיד target (location)

$\log_2 n$ מספר יונים
 $O(\log_2 n)$ מספר יונים
 $O(\log_2 n)$

16/11/18

Binary Search 2

bottom = 1

top = n

found = false

while (not found and top >= bottom)

mid = (bottom + top) / 2

if target == arr[mid].key

found = true

else if target > arr[mid].key

bottom = mid + 1

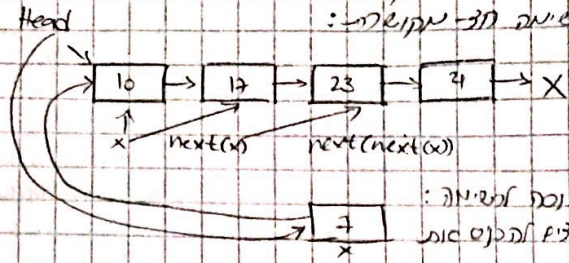
else

top = mid - 1

location = mid

ב-11/16: Binary Search 2 - 2 השאלות זכו, מספר מהשאלות במקרה הגרוע הוא $O(\log_2 n)$
במקרה הטוב, Binary 1 הוא מהיר יותר מ-Binary 2

רשימה מקושרת
- רשימה קצרה מקושרת:



- נכנסת רשימה
רשימה חדשה את

Insert (L, x)

1. $next(x) \leftarrow head(L)$
2. $head(L) \leftarrow x$

רשימה - כנסת x מחדש אל ראש הרשימה

- מחיקת רשימה:

ראשית נחזיק את כל הערכים במערך. כפי שהראינו, נכנסת רשימה. קיימת רשימה, שגורר קוצים ממחוק את 23 ואז 17 בדרך עשירי של 1-4. אי אפשר למחוק אחר כך ערך אחר, כיוון שכל הערכים יתקפו שום סדרים ממחוק.

Delete (L, x)

1. if $(x = head(L))$ then
2. $head(L) \leftarrow next(x)$
3. else
4. $y \leftarrow head(L)$
5. while $(next(y) \neq x)$ do
6. $y \leftarrow next(y)$
7. $next(y) \leftarrow next(x)$

כמתקב שרשימה ממחוק את האותיות בסדר הרשימה

רשימה
סמוכות
נחלק
כי צריך לעבור על
כל רשימה

רשימה - כל עוד y הוא לא x שאותו מחפשים ממקומות קיימת. הקוד y זה הקוד x שאותו מחפשים

- חיפוש רשימה:

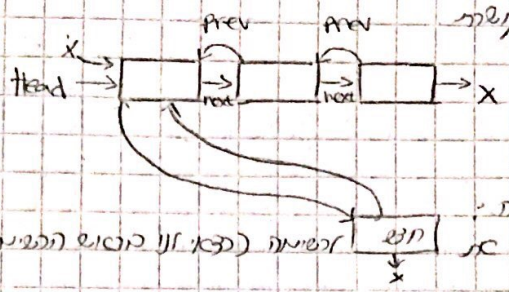
Search (L, k)

1. $x \leftarrow head(L)$
2. while $(x \neq null)$ then
3. if $(key(x) = k)$ then
4. return x
5. else
6. $x \leftarrow next(x)$
7. return null

רשימה
סמוכות
נחלק
כי צריך לעבור על
כל רשימה

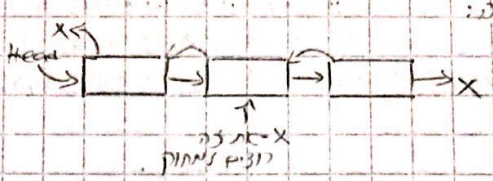
רשימה - אם התקבלה k היא ואלו נכנסו למאגר שהראשית

כדי להוסיף קובץ יצאנו את next ואת prev
 כדי להסיר קובץ יצאנו את next ואת prev



דגשן (14)
 - קובץ יצאנו את next ואת prev
 - קובץ יצאנו את next ואת prev

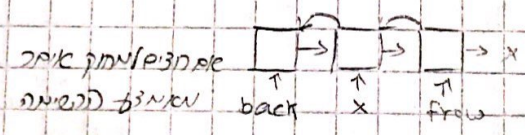
- Insert(L, x)
1. next(x) ← Head(L)
 2. prev(Head(L)) ← x
 3. prev(x) ← Null
 4. Head(L) ← x



דגשן (14)

דגשן (14)
 - קובץ יצאנו את next ואת prev
 - קובץ יצאנו את next ואת prev

- Delete(L, x):
1. if (x = Head(L)) then
 2. Head(L) ← next(x)
 3. prev(x) ← null
 4. else
 5. if (next(x) = null) then
 6. next(prev(x)) ← null
 7. else
 8. back ← prev(x)
 9. Forw ← next(x)
 10. next(back) ← Forw
 11. prev(Forw) ← back



דגשן (14)
 - קובץ יצאנו את next ואת prev
 - קובץ יצאנו את next ואת prev

L רשימה רצף מקושרת ממוינת על מספרים.

כיצד מתבצעת על L הפעולה המיועדת (חיבור, הכנסה, מחיקה), ולוודא שהיא רצף-מקושרת כלשהי (אנא מוסיף)?

- הכנסת ערך חדש:

Insert(L, x):

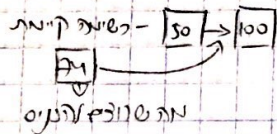
1. if (key(x) < key(Head(L))) then
2. next(x) ← Head(L)
3. Head(L) ← x
4. else
5. y ← Head(L)
6. while (key(next(y)) < key(x)) do
7. y ← next(y)
8. if (next(y) = Null) then
9. next(y) ← x
10. next(x) ← Null
11. else
12. next(x) ← next(y)
13. next(y) ← x

סיפור במקרה שבו רוצה להכניס אחרת שהיא איך שנו הכא הכי קטן ואיך אני יוצא שאת דבר זה עושים אותו זכאש והשיטה

סבוכות
O(n)
כי צריך
לדון
כל הרשמי

מקרה שבו איננו להכניסו, קודם שאני הכי קטן את והשיטה שקיימת
אם מוכנסים אותו אחרת כנראה

מקרה שבו איננו להכניסו דבר זהכא כי שני איברים קיימים ברשימה



- מחיקת ערך ברשימה:

הדיוק כמו ברשימה רצף מקושרת כלשהי!

- חיפוש ברשימה:

Search(L, k):

1. y ← head(L)
2. while (y ≠ null and key(y) ≤ k) do
3. if (key(y) = k) then
4. return y
5. else
6. y ← next(y)
7. return null

B או C מיועדת צדד קטן/ארוך מהם שאינם מחסלים אישית האחד.

סבוכות
O(n)

קבלת גודל הריבועים בסדרה (המשך):
 * סדרה מתוארת: a_n און הריבועים
 * סדרה מתוארת הריבועים: a_n און הריבועים

- Q_1, Q_2, \dots, Q_n
- לגוד Q_n : n הריבועים
- לגוד Q_{n-1} : $n-1$ הריבועים
- ...
- לגוד Q_3 : 2 הריבועים
- לגוד Q_2 : 1 הריבועים
- לגוד Q_1 : 0 הריבועים

$$1+2+\dots+(n-1) = \frac{n(n-1)}{2} = \frac{n^2}{2} - \frac{n}{2} \approx \frac{n^2}{2}$$

גודל הריבועים הוא $O(n^2)$ און, גודל הריבועים הוא $O(n^2)$

קאזואליזציה מנוסחת (המשך): גודל הריבועים און הריבועים און הריבועים און הריבועים

$$\{1, 2, \dots, n-1, n\} \leftarrow \text{המשך בין הריבועים און הריבועים}$$

$$\frac{(n-3)(n-2)}{2}$$

$$\frac{(n-2)(n-1)}{2} - \frac{(n-3)(n-2)}{2} = \frac{(n^2 - 3n + 2) - (n^2 - 5n + 6)}{2} = \frac{2n - 4}{2} = n - 2$$

$$\frac{4n-4}{2} = 2n-2$$

:(Bubble-Sort)

$$[2, 3, 1, 5, 0, 7, 9] \leftarrow [2, 3, 1, 5, 0, 9, 7] \leftarrow [2, 3, 5, 1, 7, 0, 9] \leftarrow [2, 5, 3, 1, 7, 0, 9] \leftarrow [5, 3, 3, 1, 7, 0, 9]$$

אנטי-סימטריה און הריבועים און הריבועים און הריבועים און הריבועים

$$[2, 1, 3, 0, 5, 7, 9] \leftarrow [2, 1, 3, 0, 5, 9, 7] \leftarrow [2, 1, 3, 5, 0, 7, 9] \leftarrow [2, 3, 1, 5, 0, 7, 9]$$

אנטי-סימטריה און הריבועים און הריבועים און הריבועים און הריבועים

$$[1, 2, 0, 3, 5, 7, 9] \leftarrow [1, 2, 3, 0, 5, 7, 9] \leftarrow [2, 1, 3, 0, 5, 7, 9]$$

$$[1, 0, 3, 5, 7, 9] \leftarrow [1, 2, 0, 3, 5, 7, 9]$$

$$[0, 1, 2, 3, 5, 7, 9] \leftarrow [0, 1, 2, 3, 5, 7, 9] \leftarrow [1, 0, 2, 3, 5, 7, 9]$$

גודל הריבועים און הריבועים און הריבועים און הריבועים
 $c = \frac{1}{2}$, $O(n^2)$ און הריבועים און הריבועים

23/11/18

אולי כשנת:

```

For i = n down to 2:
  for j = 2 to i
    if A[j-1] > A[j]
      exchange(j-1, j)

```

אולי כשנת:

```

For j = 2 to i
  if A[j-1] > A[j]
    exchange(j-1, j)

```

אולי:

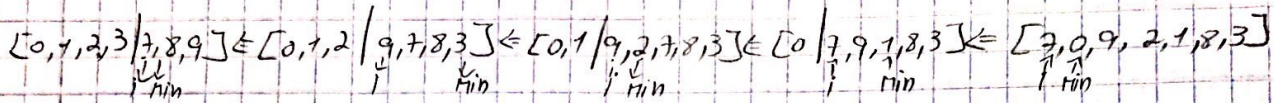
```

A[j-1] > A[j]
exchange(j-1, j)
T = A[j-1]
A[j-1] = A[j]
A[j] = T

```

מיון סלקציה (Selection-Sort)

מיון סלקציה זה מתבצע על ידי בחירת האיבר הקטן ביותר במערך ובחילופו עם האיבר הראשון במערך. תהליך זה מתבצע עד שהמערך מיון.



מיון סלקציה: $[0, 1, 2, 3, 7, 8, 9]$

```

For i = 1 to n-1
  i_min = i
  if A[j] < A[i_min]
    i_min = j
  if i < i_min
    exchange(i, i_min)

```

מספר השוואות:

- n-1
- n-2
- ...
- 1

$$\frac{n(n-1)}{2} \approx \frac{n^2}{2} = O(n^2)$$

מיון סלקציה הוא מיון סלקציה. מספר השוואות הוא $O(n^2)$.

מיון - מיון סלקציה זה מתבצע על ידי בחירת האיבר הקטן ביותר במערך ובחילופו עם האיבר הראשון במערך.

- 1- ארבעה תלמידי
- 10- תלמידי "גילי"
- 20- תלמידי "מאיר"
- 1- תלמידי "קובי"
- 2- תלמידי "דניאל"

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150

: 80 תלמידי (

X $A[13] = 80, i=1$ -10
 X $A[23] = 80, i=2$
 X $A[33] = 80, i=3$
 ...
 X $A[73] = 80, i=7$
 ✓ $A[83] = 80, i=8$

↑
 "גילי"
 8 תלמידי

ד- קבוצת כל ה-15 (הכאן והכאן) $\Rightarrow b=1, t=15$ -2

(מחלקים את ה-15 תלמידי) $\Rightarrow M = \left\lfloor \frac{1+15}{2} \right\rfloor = 8$

(קובעים את המספר החדש, כלומר, כל מי שהמספר שלו < 8) $\Rightarrow X = A[83] < 80$

מחלקים את ה-8 תלמידי $\Rightarrow b=8, t=15$

$M = \left\lfloor \frac{8+15}{2} \right\rfloor = 11$

(קובעים את המספר החדש, כלומר, כל מי שהמספר שלו < 8) $\Rightarrow X = A[113] < 80$

מחלקים את ה-11 תלמידי $\Rightarrow b=8, t=11$

$M = \left\lfloor \frac{11+8}{2} \right\rfloor = 9$

(קובעים את המספר החדש, כלומר, כל מי שהמספר שלו < 8) $\Rightarrow X = A[93] < 80$

מחלקים את ה-9 תלמידי $\Rightarrow b=8, t=9$

✓ $A[83] = 80$

$b=t$

↑
 "גילי"
 8 תלמידי

ה-15 תלמידי (הכאן והכאן) $\Rightarrow b=1, t=15$ -3

$M = \left\lfloor \frac{1+15}{2} \right\rfloor = 8$

✓ $A[83] = 80$

↑
 "גילי"
 8 תלמידי

צבועים על המעגלים א' ו-ב
א' ו-ב

המשפט
המשפט 2
 $X A [3] = 90, i=1 - 1$
 \vdots
 $X A [8] = 90, i=8$
 $\sqrt{A [9]} = 90, i=9$
 \Downarrow
משפט 1 - משפט 1

ה- סעיף 10

המשפט 1 - המשפט 1
 $b=1, t=15$
 $M = \left[\frac{1+15}{2} \right] = 8$

$\sqrt{A [8]} < 90$ - הוכחה כי המשפט 1 נכון.
משפט 1 - המשפט 1
 $b=9, t=15$
 $M = \left[\frac{9+15}{2} \right] = 12$

$X A [12] < 90$ - הוכחה כי המשפט 1 נכון.
משפט 1 - המשפט 1
 $b=9, t=12$
 $M = \left[\frac{9+12}{2} \right] = 10$

$X A [10] < 90$ - הוכחה כי המשפט 1 נכון.
משפט 1 - המשפט 1
 $b=9, t=10$
 $M = \left[\frac{9+10}{2} \right] = 9$

$X A [9] < 90$ - הוכחה כי המשפט 1 נכון.
משפט 1 - המשפט 1
 $b=9, t=9$
 $\sqrt{A [9]} = 90$
 \Downarrow

משפט 5 - המשפט 5

המשפט 1 - המשפט 1
 $b=1, t=15$
 $M = \left[\frac{1+15}{2} \right] = 8$
 $X A [8] = 90$
 $\sqrt{A [8]} < 90$
 $b=9, t=15$
 $M = \left[\frac{9+15}{2} \right] = 12$

$X A [12] = 90$
 $X A [12] < 90$
 $b=9, t=12$
 $M = \left[\frac{9+12}{2} \right] = 10$

$X A [10] = 90$
 $A [10] < 90$
 $b=9, t=10$
 $\sqrt{A [9]} = 90$
 \Downarrow

משפט 5 - המשפט 5

$$\begin{aligned}
 &: 25 \text{ זמן } (3) \\
 & \times A[3] = 25, i=1 \quad -10 \\
 & \times A[2] = 25, i=2 \\
 & \vdots \\
 & \times A[1] = 25, i=5 \\
 & \Downarrow \\
 & \text{כל זמן}
 \end{aligned}$$

הזמן הוא 30 כל 3 זמן כי כל פעם נקרא את כל האיברים ויש 30 איברים

$$\begin{aligned}
 & \times A[1] = 25, i=1 \quad -2 \\
 & \times A[2] = 25, i=2 \\
 & \times A[3] = 25, i=3 \\
 & \Downarrow \\
 & \text{כל זמן } 3
 \end{aligned}$$

הזמן הוא 3 כל זמן

$$b=1, t=15 \quad -1$$

$$M = \left\lfloor \frac{1+15}{2} \right\rfloor = 8$$

$$\times A[8] < 25$$

$$b=1, t=8$$

$$M = \left\lfloor \frac{1+8}{2} \right\rfloor = 4$$

$$\times A[4] < 25$$

$$b=1, t=4$$

$$M = \left\lfloor \frac{1+4}{2} \right\rfloor = 2$$

$$\sqrt{A[2]} < 25$$

$$b=3, t=4$$

$$M = \left\lfloor \frac{3+4}{2} \right\rfloor = 3$$

$$\times A[3] = 25$$

$$b=3, t=3$$

$$\times A[3] = 25$$

↓

הזמן הוא 5 כל זמן

$$b=1, t=15 \quad -3$$

$$M = \left\lfloor \frac{1+15}{2} \right\rfloor = 8$$

$$\times A[8] = 25$$

$$\times A[8] < 25$$

$$b=1, t=7$$

$$M = \left\lfloor \frac{1+7}{2} \right\rfloor = 4$$

$$\times A[4] = 25$$

$$\times A[4] < 25$$

$$b=1, t=3$$

$$M = \left\lfloor \frac{1+3}{2} \right\rfloor = 2$$

$$\times A[2] = 25$$

$$\sqrt{A[2]} < 25$$

$$b=3, t=3$$

$$\times A[3] = 25$$

↓

הזמן הוא 7 כל זמן

הזמן הוא 7 כי כל פעם נקרא את כל האיברים ויש 7 איברים

3/11/18

134-230

200
137

$O(n)$ -1

$O(n)$ -2

$O(\log_2 n)$ -2

$O(\log_2 n)$ -3

Optimal $O(n^2)$ $\left\{ \begin{array}{l} \text{מיון קבוע} \\ \text{מיון דינמי} \end{array} \right.$

אנדרטיות של מקרים (קיימות רק לרובות עקב כמות מקרים)
המיון (מלא) יחסית לזמן המיון
מיון דינמי אינו מיון חיצוני (אולי)

Insertion - Sort

$\leftarrow [2, 3, 4, 7 | 1, 6] \leftarrow [2, 4, 7 | 3, 1, 6] \leftarrow [4, 7, 2, 3, 1, 6] \leftarrow [4, 3, 2, 3, 1, 6]$
 $[1, 2, 3, 4, 6, 7] \leftarrow [1, 2, 3, 4, 7 | 6]$

מיון קבוע $[a_1, a_2, \dots, a_n]$

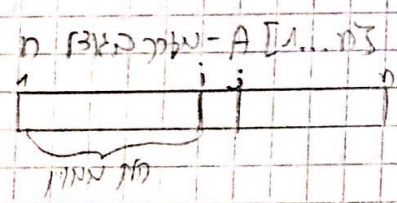
- $a_1 \leq a_2$: אנושני
- $a_1 > a_2$: הקטן עם זמן a_1
- $a_3 \geq a_2$: אנושני
- $a_3 < a_2$: מניח 3 עם a_2
- $a_3 \geq a_1$: מניח 3 עם a_1
- $a_3 < a_1$: מניח 3 עם a_1

$b=T, a=b, T=a$ $[1, 2, 3]$ - מיון קבוע, אנושני
 a, b : המספר של 2 - $[1, 3, 2]$

$c=T, b=c, a=b, T=a$ a, b, c : המספר של 3 - $[3, 1, 2]$

$[1, 2, 3] \leftarrow [1, 2, 3] \leftarrow [1, 3, 2] \leftarrow [3, 1, 2]$ (המספר של 3)

$[1, 2, 3] \leftarrow [2, 3, 1] \leftarrow [3, 2, 1]$ (המספר של 3)



```

INSERTION-SORT (A)
for j = 2 to n
  i = j - 1; t = A[j]
  while (i > 0 and A[i] > t)
    A[i+1] = A[i]; i = i - 1
  A[i+1] = t
  
```

סדרת המספרים של איברים *
 $j=2$: מספרת 1 : המספרת *
 $j=3$: מספרת 2 : המספרת *
 $j=4$: מספרת 3 : המספרת *
 $1+2+\dots+(n-1) = \frac{n(n-1)}{2} = O(n^2)$

מיון דינמי יקוי יחסית מיון קבוע ולא מיון דינמי.

מיון דינמי יחסית מיון קבוע מיון דינמי מיון דינמי

מיון דינמי יחסית מיון קבוע מיון דינמי מיון דינמי

מיון דינמי יחסית מיון קבוע מיון דינמי מיון דינמי

A[1...n]

$1 \leq p \leq r \leq n$
 A[p...r] : מערך

Merge Sort (A, p, r)

$q = \lfloor (p+r)/2 \rfloor$

Merge Sort (A, p, q)
 Merge Sort (A, q+1, r)
 Merge Sort (A, p, q, r)

[7, 9, 5, 2, 12, 8, 6]

[1, 15, 13, 11, 8, 10]

[2, 5, 6, 7, 8, 9, 12]

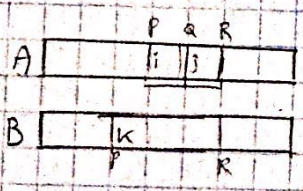
[1, 8, 9, 10, 13, 15]

[1, 2, 5, 6, 7, 8, 8, 9, 9, 10, 12, 13, 15]

באמצעות שני המערך
 ונקודות את המספרים
 הממוינים.
 את המערך B
 ו-15 של המערך A
 הממוינים בנקודות
 המספרים הממוינים
 את המערך B

המערך B
 הממוינים בנקודות
 המספרים הממוינים
 את המערך B
 ו-15 של המערך A
 הממוינים בנקודות
 המספרים הממוינים
 את המערך B

$O(n \log n)$
 $O(n)$



```

Merge(A, p, q, r)
i = p      k = p
j = q + 1
while (i <= q and j <= r)
    if (A[i] <= A[j])
        B[k] = A[i]
        k = k + 1
        i = i + 1
    else
        B[k] = A[j]
        k = k + 1
        j = j + 1
while (i <= q)
    B[k] = A[i]
    k = k + 1
    i = i + 1
while (j <= r)
    B[k] = A[j]
    k = k + 1
    j = j + 1
    
```


30/11/18

$$\text{Mid} = \left\lfloor \frac{b+t}{3} \right\rfloor$$

ה' 2 נקודות עמודים 2/6

4.2.1.2.2

1	2	3	4	5	6	7	8	9	10	11	12	
A	10	20	30	40	50	60	70	80	90	100	110	120

: A -> 12 עמודים - 12
b=1, t=12

$$\text{Mid} = \left\lfloor \frac{1+12}{3} \right\rfloor = 4$$

x A[4] = 60
√ A[4] < 60

b=5, t=12

$$\text{Mid} = \left\lfloor \frac{5+12}{3} \right\rfloor = 5$$

x A[5] = 60
√ A[5] < 60

b=6, t=12

$$\text{Mid} = \left\lfloor \frac{6+12}{3} \right\rfloor = 6$$

√ A[6] = 60
↓

מיקום 5, 12 נכון

: A -> 110 עמודים - 12

b=1 t=12

$$\text{Mid} = \left\lfloor \frac{1+12}{3} \right\rfloor = 4$$

x A[4] = 110
√ A[4] < 110

b=5, t=12

$$\text{Mid} = \left\lfloor \frac{5+12}{3} \right\rfloor = 5$$

x [5] = 110
√ [5] < 110

b=6, t=12

$$\text{Mid} = \left\lfloor \frac{6+12}{3} \right\rfloor = 6$$

x A[6] = 110
√ A[6] < 110

b=7, t=12

$$\text{Mid} = \left\lfloor \frac{7+12}{3} \right\rfloor = 6$$

x A[6] = 110
√ A[6] < 110

ה' 2 נקודות עמודים 2/6 - b=7, t=12

: A = 5, t = 12
 b = 1, t = 12
 Mid = $\lfloor \frac{12-1}{3} \rfloor + 1 = 4$
 x A [4] = 60
 ✓ A [4] < 60

b = 5, t = 12
 Mid = $\lfloor \frac{12-5}{3} \rfloor + 5 = 7$
 x A [7] = 60
 ✓ A [7] < 60

→ מיון ממוצע

b = 5, t = 6
 Mid = $\lfloor \frac{6-5}{3} \rfloor + 5 = 5$
 x A [5] = 60
 ✓ A [5] < 60

b = 6, t = 6
 ✓ A [6] = 60
 ↓

מיון ממוצע

: A = 110, t = 12
 b = 1, t = 12
 Mid = $\lfloor \frac{12-1}{3} \rfloor + 1 = 4$
 x A [4] = 110
 ✓ A [4] < 110

b = 5, t = 12
 Mid = $\lfloor \frac{12-5}{3} \rfloor + 5 = 7$
 x A [7] = 110
 ✓ A [7] < 110

b = 8, t = 12
 Mid = $\lfloor \frac{12-8}{3} \rfloor + 8 = 9$
 x A [9] = 110
 ✓ A [9] < 110

b = 10, t = 12
 Mid = $\lfloor \frac{12-10}{3} \rfloor + 10 = 10$
 x A [10] = 110
 ✓ A [10] < 110

b = 11, t = 12
 Mid = $\lfloor \frac{12-11}{3} \rfloor + 11 = 11$
 ✓ A [11] = 110
 ↓

מיון ממוצע 9, 10, 11

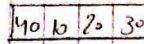
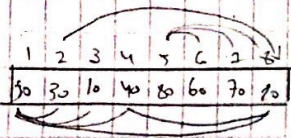
x סיבוכיות מיישם בריבועים

$O(\log n)$

$$\left(\begin{matrix} h \cdot \left(\frac{2}{3}\right)^i - 1 \\ n = \left(\frac{2}{3}\right)^i \\ i = \log_{1.5} n \end{matrix} \right)$$

מיון

* היסוד במספר: כל איברי המערך הם מספרים טבעיים



3 הפוסקים

40 זכר ו-40 נקבה

לפי סדר המספרים

- הפוסקים 4
 - הזכרים 2
 - הנקבות 1
 - הזכרים 3
 - הנקבות 1
 - הזכרים 1
- } 12 הפוסקים

* המיון מתבצע על ידי הפוסקים! סדר המיון

1) לראות מהו סדר המיון

א- מספר הפוסקים המקסימלי

$$\frac{n(n-1)(n-2) \dots 1}{2} = \frac{n(n-1)}{2}$$

ב- מספר הפוסקים המינימלי אפשרי
כאשר המספר מינימלי 0

2) ניתן לקבוע את מספר הפוסקים:

- א- זכרים ונקבות וההפרש מספר הפוסקים ב-A
- ב- מספר הפוסקים זכרים ונקבות
- ג- מספר הפוסקים זכרים ונקבות
- ד- מספר הפוסקים זכרים ונקבות

Inversion (A, n)

1. Counter ← 0
2. for i ← 1 to n-1
3. for j ← i+1 to n
4. if A[j] < A[i] then
5. Counter ← Counter + 1
6. return (Counter)

סכומים:

מספר הפוסקים המקסימלי:

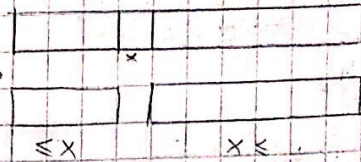
מספר הפוסקים המקסימלי	מספר i
n-1	1
n-2	2
n-3	3
⋮	⋮
1	n-1

$$\frac{(n-1) + (n-2) + \dots + 1}{2} = \frac{n(n-1)}{2} = O(n^2)$$

* המספרים הנכונים (המספרים הנכונים) הם $O(n \log n)$

Quick Sort - מיון מהיר

(PARTITION) חלוקה →



מיון מהיר $A[1..n]$ Hoare

$x = A[p]$ מיון מהיר $A[p..r]$ Lomuto

$A[p..q]$, $A[q+1..r]$ מיון מהיר

מיון מהיר $A[p..r]$ (מיון מהיר) - $O(N \log N)$

Quick Sort (A) מיון מהיר

Quick Sort (A, p, r) מיון מהיר

```

if (p < r)
    q = partition(A, p, r)
    Quick Sort (A, p, q)
    Quick Sort (A, q+1, r)
    
```

partition(A, p, r) =

```

x = A[p]
i = p-1
j = r+1
while true
    do j = j-1
    while A[j] > x
    do i = i+1
    while A[i] < x
    if i < j
        exchange A[i] ↔ A[j]
    else return j
    
```

$A = [13, 19, 9, 5, 12, 8, 7, 4, 11, 2, 6, 21]$

partition(A, 1, 12)

$p=1, r=12$
 $j=11, A[j]=6$
 $i=1, A[i]=13$

$A[1] \leftrightarrow A[11]$

$A = [6, 19, 9, 5, 12, 8, 7, 4, 11, 2, 13, 21]$
 $19 > 13 > 2$

$A = [6, 2, 9, 5, 12, 8, 7, 4, 11, 19, 13, 21]$

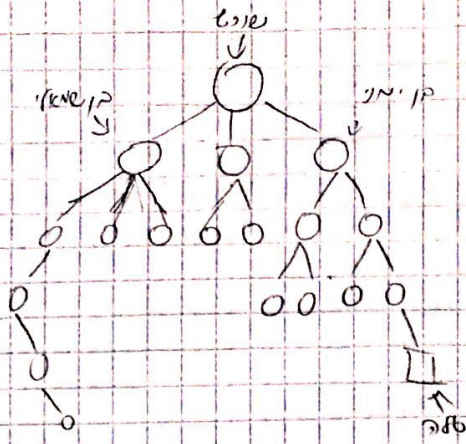
$j=9, i=10, i > j$ - return $j=9 \rightarrow q=9$
 $q+1=10$

$A_1 = [6, 2, 9, 5, 12, 8, 7, 4, 11]$

$A_2 = [19, 13, 21]$

TREES - עץ

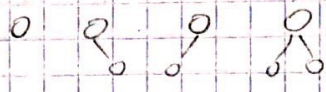
- parent = אב
- Root = שורש
- children = ילדים
- Node = נוד
- LEAF = עלה
- key = מפתח



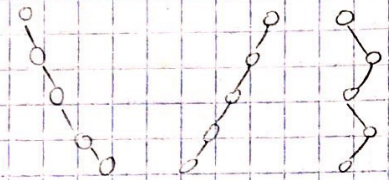
עץ הוא מבנה נתונים, אשר מורכב מנודים, הנקראים ילדים, הנקראים קבוצה.

עץ בינארי

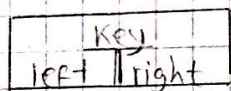
עץ בינארי הוא עץ שבו לכל נוד יש לכל היותר שני ילדים.



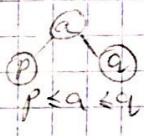
העץ הוא מבנה נתונים.



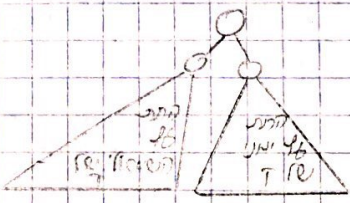
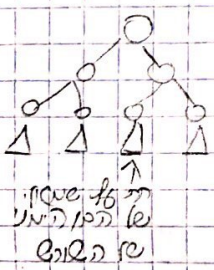
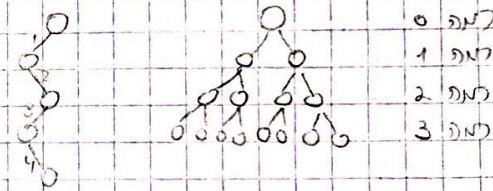
קיימים שני סוגים של עץ בינארי: עץ מלא ו-עץ רגיל.



העץ הוא מבנה נתונים, אשר מורכב מנודים, הנקראים ילדים, הנקראים קבוצה.

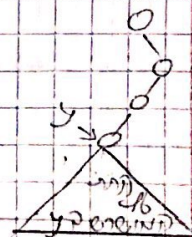
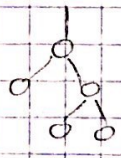
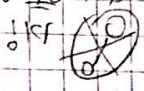


העץ הוא מבנה נתונים, אשר מורכב מנודים, הנקראים ילדים, הנקראים קבוצה.



עץ בינארי

(Full tree) עץ מלא הוא עץ שבו לכל נוד יש לכל היותר שני ילדים.



עץ בינארי

עץ בינארי הוא עץ שבו לכל נוד יש לכל היותר שני ילדים.



עץ בינארי

$2^0 + 2^1 + \dots + 2^{n-1} = 2^n - 1$

עץ בינארי הוא עץ שבו לכל נוד יש לכל היותר שני ילדים.

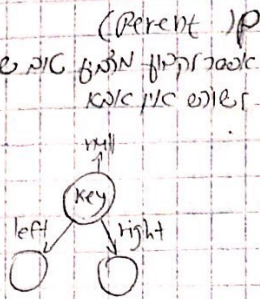
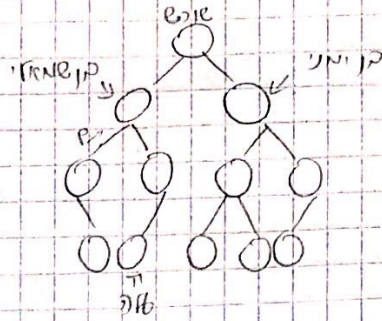
14/12/18

צד חיפוש בינארי

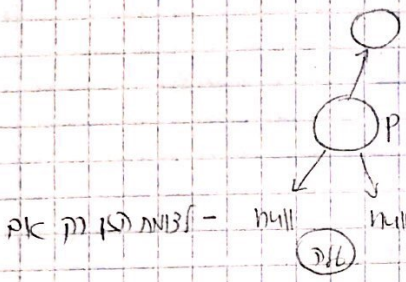
Search = חיפוש
Insertion = הכנסה
Deletion = מחיקה

ט/ווח'א/א/א:
הקנסה חייב
מחיקה חייב

ח: אם הפרטים, אם המבנה (חשוב שיהיה ניתן להשווה בין המבנה)



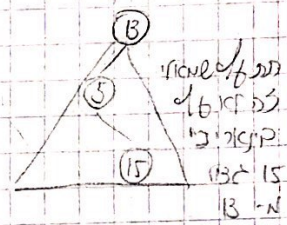
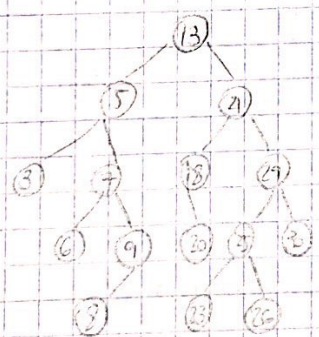
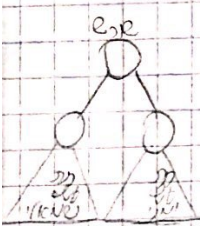
(Parent) P
אם יש קישור מאלו
זוהי אבן אבן



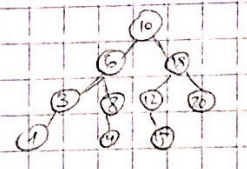
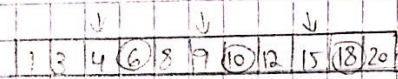
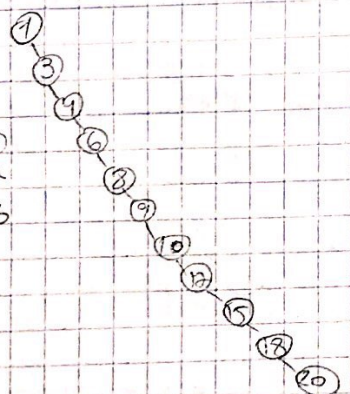
אזכור - פירוט נקודות מסוימות

חיפוש - הוא על ידי שימוש בשיטת חיפוש בינארית.

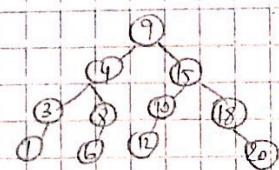
* כל מספר קטן מהאבא קטן מן המספרים
* כל מספר קטן מהאבא גדול מן המספרים



כדי להבין את המבנה
חייבים להבין את המבנה
על ידי שימוש במחשבים



חיפוש בינארי




חיפוש בינארי
* המבנה הוא חייב להיות יעיל


14/12/22

מספר 16

גיוון זמני - סיבוכיות:

קבוצה - מנהל של פאריס במלך, על כוס מספר אחיד. קבוצה זמנית.

$O(n)$ 
 $O(n+(n-1)+(n-2)+\dots+1) = O\left(\frac{n(n+1)}{2}\right) = O(n^2)$
 קבוצה זמנית - סיבוכיות: $O(n^2)$

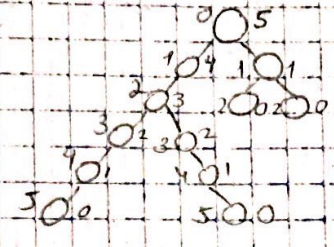
$O(n)$ 
 $O(n) \cdot \log_2 n = O(n \log n)$
 סיבוכיות זמנית:



1.6.1 קישורים - זכרון של צמתים

X צמת

- 1. left(x) - ימין
- 2. right(x) - שמאל
- 3. parent(x) - הורה
- 4. isLeaf(x) - האם צומת היא עלה
- 5. isRoot(x) - האם צומת היא שורש
- 6. isNode(x) - האם צומת היא צומת (מחזורי)
- 7. isEdge(x) - האם צומת היא קצה
- 8. isParent(x) - האם צומת היא הורה



אם יש ימין - ימין
אם יש שמאל - שמאל

1.6.2 האם צומת היא שורש? האם צומת היא עלה? האם צומת היא הורה?

Root(x)

1. if (parent(x) == null) true
2. return true
3. else
4. return false

DEPTH(x)

1. if (root(x) == true) true
2. return 0
3. else
4. return (1 + DEPTH(parent(x)))

Height(x)

1. if (left(x) == null and right(x) == null) then
2. return 0
3. if (left(x) == null) then
4. return (1 + Height(right(x)))
5. if (right(x) == null) then
6. return (1 + Height(left(x)))
7. return 1 + Max(Height(left(x)), Height(right(x)))

* פונקציה שמחזירה את גובה הצומת

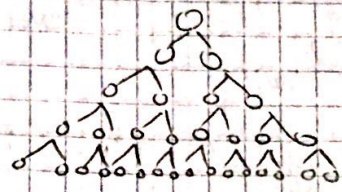
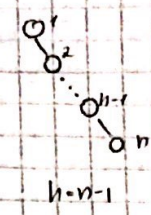
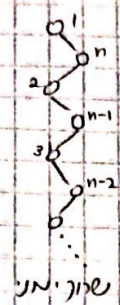
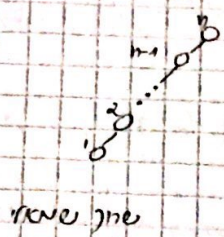
21/12/18

$$\log_2 n < h < n$$

↓ ↓

מספר קומות מספר קומות

מספר קומות - מספר קומות



$$2^h \leq n < 2^{h+1}$$

$$h \approx \log_2 n$$

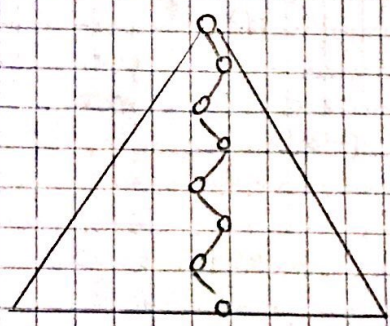
מספר קומות

```

Tree-Search(x, k)
if (x = null or k = x->key)
return x
if (x < x->key)
return Tree-Search(x->left, k)
else
return Tree-Search(x->right, k)

```

מספר קומות



$O(h)$: מספר קומות
 $O(\log n)$: מספר קומות
 $O(n)$: מספר קומות

* מציאת המקסימום
מציאת המינימום במסערת המקסימום בתת-עץ המיושם ב-x

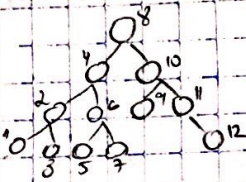
```

O(h) {
    Tree-Maximum(x)
    while (x->right != null)
        x = x->right
    return x
}
    
```

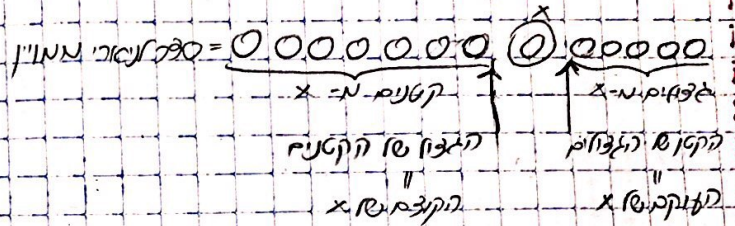
* מציאת המקסימום
מציאת המינימום במסערת המינימום בתת-עץ המיושם ב-x

```

O(h) {
    Tree-Minimum(x)
    while (x->left != null)
        x = x->left
    return x
}
    
```



Successor : אפ"ב
Predecessor : קודם



אם יש ימין של x אז המקסימום של הימין הוא ה-Successor.
אם אין ימין של x אז המקסימום של הימין הוא ה-Predecessor.

O(h)

```

Tree-Successor(x)
if (x->right != null)
    return Tree-Minimum(x->right)
y = x->p
while (y != null and x == y->right)
    x = y
    y = y->p
return y
    
```

O(h)

```

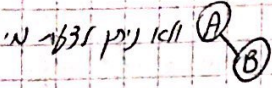
Tree-Predecessor(x)
if (x->left != null)
    return Tree-Maximum(x->left)
y = x->p
while (y != null and x == y->left)
    x = y
    y = y->p
return y
    
```


קלף חיבטט ביני (קלף - BST)
* סדר תחזי: צמת, קן שמאלי, קן ימני
x סדר תמי: קן שמאלי, צמת, קן ימני
x סדר סוסי: קן שמאלי, קן ימני, צמת

1) הדינמן בשמאלות של צמתים בסדר סדרה ירון, האם ניתן לשחזר את הקלף משוקה?

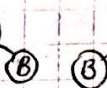
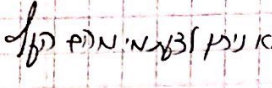
סדר תחזי - לא ניתן.

לקיבו השמאלה בסדר תחזי B, A מתאימים הקלפים
האם הקלף



סדר תמי - לא ניתן.

לקיבו השמאלה בסדר תמי B, A מתאימים הקלפים
האם הקלף



סדר סוסי - לא ניתן.

לקיבו השמאלה בסדר סוסי B, A מתאימים הקלפים
האם הקלף

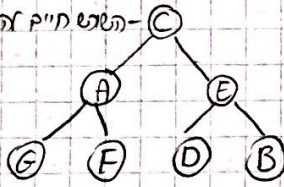


2) הדינמן 2 השמות סדר תחזי וסדר תמי, האם ניתן לשחזר את הקלף?

זרימא שמת שנית:

סדר תחזי - B, D, E, F, G, A, C
סדר תמי - B, E, D, C | F, A, G

השש חייב ארוך לפי סדר תחזי.



ניתן!

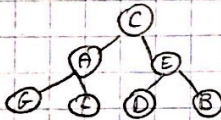
העק הכאשון בסדר תחזי תמיד זהות בשונה הקלף
לאחר מציאת השורש, ברשימה בסדר תמי, כן נני
שולטנו נמצא קתת קלף שמאלי של השורש.

ד מ' שמאלו נמצא קתת קלף ימני של השורש. ברמסאים קתת קלף שמאלי וימני לחוב, וקר האור

3) הדינמן 2 השמות סדר סוסי וסדר תמי, האם ניתן לשחזר את הקלף?

אפי הקלף השאלור הקורמת נכחית את הסדר הסוסי.

סדר סוסי - C, E, B, D, A, F, G
סדר תמי - B, E, D, C, F, A, G



ניתן!

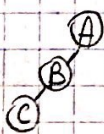
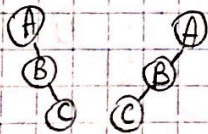
כמו השאלור הקורמת, רק שמאלה שורש הקלף
מצבים העק הסדרון ברשימה בסדר סוסי.

4) הדינמן 2 השמות סדר תחזי וסדר סוסי, האם ניתן לשחזר את הקלף?

סדר תחזי - B, D, E, F, G, A, C
סדר סוסי - C, E, B, D, A, F, G

לא ניתן!

לקיבו השמות בסדר תחזי A, B, C וסדר סוסי A, B, C מתאימים הקלפים
ולא ניתן אצגמ מ' מהם הקלף.

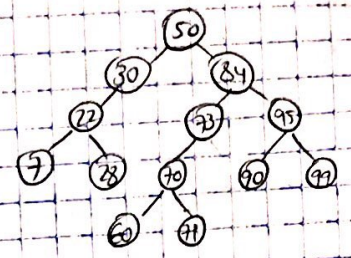


כל חיפוש בינארי: כל קיני שמתחת לדבר מספרים
 מסודר אפי-מבנה כל חיפוש בינארי, הזמרה-לכבוד כל צימת בתת-כלל שמאזי שאל
 יש נייטב קטע או שווים או, כהת-כלל יתנו שאל יש (תכום סגורים או שנויים או)

אצמא - תרגום: 10, 20, 30



(מן הקלה, אונחני תרגום אצמא עם קיני בו עוקר - 71? (המיתנו, ולא רואים את צב הקלון)

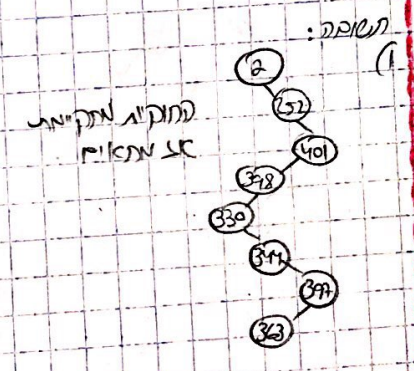
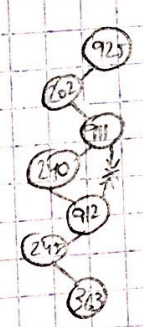


המחשבים קטנים ושהג: כה-שם השווים עק שמתחם אלוהם, אה הוא שווה-מזכאו. אם אינע שווה, אפ טעשים שמאזי
 כאשר קטן מהשוש וימנה כאשר-אצלו מהשוש אק פואה. אם טעשים אצמא - עק זא מנזאו.
 סימכות: $h \leq \log_2 n$ ו- $h \leq \log_2 n$. (ה) ג- אהת קלל C

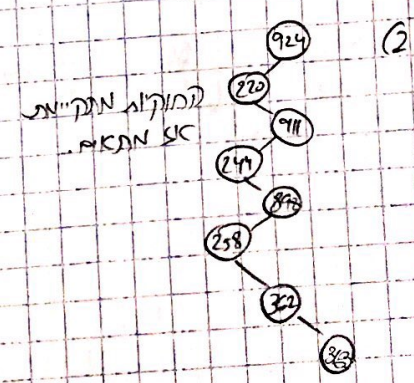
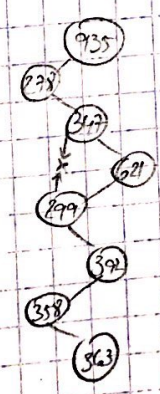
כל חיפוש בינארי מאוחסני לביים שזמין קן 1 - 1000 מחשבים קלל אור תרגום 363
 אלו מהשמות הכסומות זמיות ספרות של לביים שנתקבלו בחיפוש?

- 1) 2, 272, 401, 398, 330, 344, 397, 363
- 2) 924, 220, 911, 244, 898, 258, 362, 363
- 3) 925, 202, 912, 240, 911, 202, 925, 363
- 4) 2, 399, 387, 219, 266, 382, 381, 278, 363
- 5) 935, 278, 347, 621, 299, 392, 358, 363

קללים (מזכאת ק- 912, גלזמ- וויי אלו קצב שמאזי שאל, עשנ את התבונה זקן וא מנזאו



המיתנו (מזכאת ק- 299, קטן מ- 347 ונמזאו קצב שאל, 116

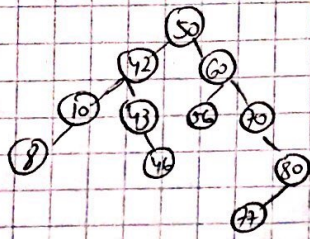


21/12/18

חשבון
מחשב

סדרה של מספרים היא חזרה ביניהם מספרים (ניתן סדרה מתחילה)

80, 77, 70, 60, 56, 50, 46, 43, 42, 10, 8 ←



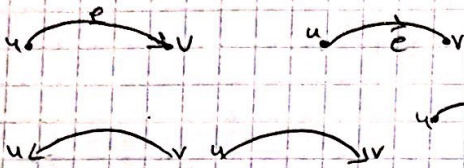
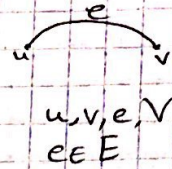
אלגוריתמים

אלגוריתמים בגרפים

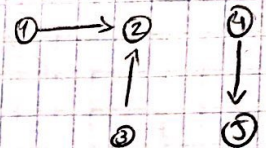
$G=(V,E)$ גרף
 $V = \text{vertex}$ צמת
 $E = \text{edge}$ קשת
 גרף סוס - קבוצה סוסית V

* תמיד בגרף מסווג יפה תל
 אם הקשת \rightarrow

$n = |V|$ מס' הצמתים n - $V = (v_1, v_2, \dots, v_n)$



גרף מכוון - n צמתים, n קשתות
 גרף קובא מכוון - אם יש קשת מכוונת
 גרף קשת מכוון - אם יש קשת קשת מכוונת
 קשת קשת מכוונת - שקורה גרף של קשתות מכוונות
 אולאה קשתית - $u \rightarrow u$



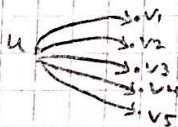
שני רכסות קשתות $V = \{1, 2, 3, 4, 5\}$ צמת
 $E = \{(1,2), (1,3), (3,2), (4,5)\}$ קשתות

גרף מכוון
 - דוקשת e יוצאת n ונכנסת v
 v סמוך u , u איש סמוך v (הש סוכי)
 וצמת v סמוך צמת u EDJACENT

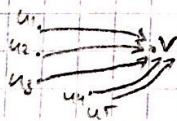
קשת קשת מכוון u סמוך v ו v סמוך u

גרף מכוון $G=(V,E)$

$n=5$ ו 4 צמתים סמוכים, אינרים, סבבית קינאט של u היא 4 .

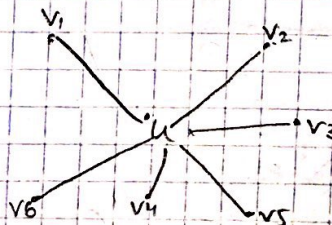


קשת v סמוך 5 צמתים אחרים, אינרים, סבבית קינאט של v קשת 5 .

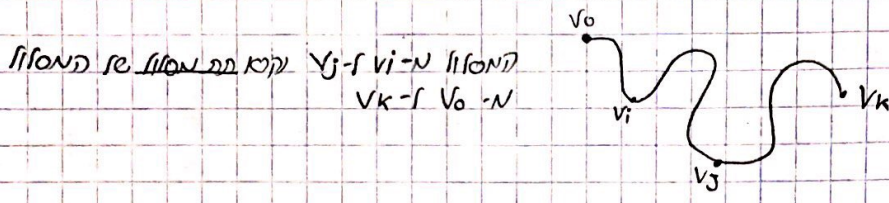
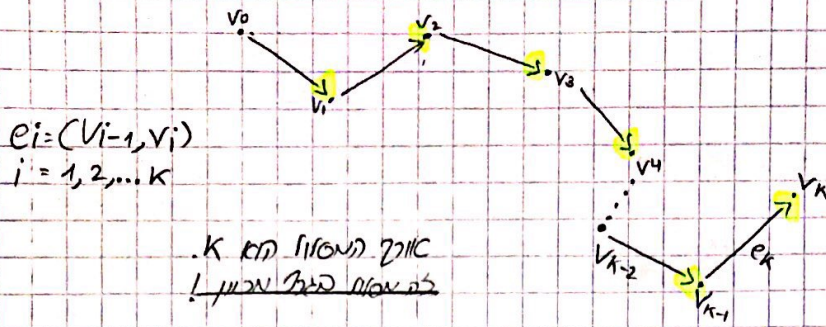


סבבית קשת מכוון:

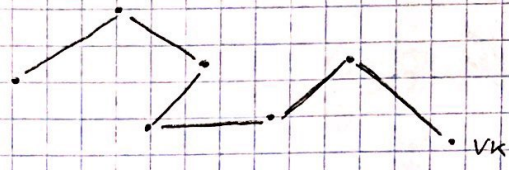
צמת u ו 6 צמתים סמוכים, קשת u קשת 6 .



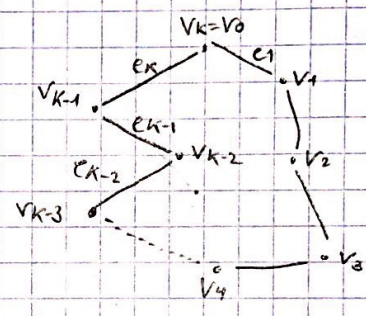
הסתובב בלולאה: מסלול מורכב מ- $(k+1)$ צמתים מתוקפים על ידי k קשתות



באופן צומת, העיבוד מסלול סגור סגור-מסלול
 אלטרנטיבי זה שאין חזרים בסלול הקשוח

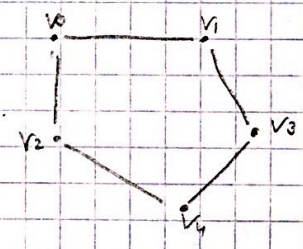


מסלול סגור = מסלול
 במקרה של מסלול הנוצרת הסלול
 עם צמתים
 אורך המסלול הוא k
 עם הצמתים למה זאת הקשוח

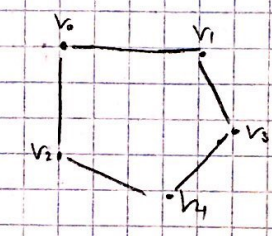


V צומת עם צבע \emptyset (סומם, אין צבע) (קרא צומת מבודד ISOLATED)

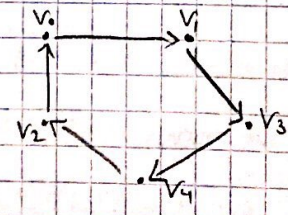
מסלול סגור סגור-מסלול, (אין חזרים) למה מסלול.



מסלול סגור סגור-מסלול, קרא ואת מסלול סגור-מסלול (כיוון זה כיוון זה מסלול
 מסלול סגור)



מסלול סגור סגור-מסלול (כיוון זה כיוון זה מסלול
 מסלול סגור)

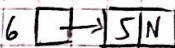
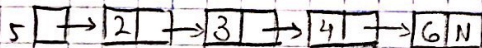
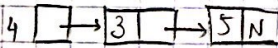
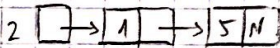
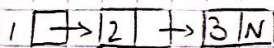
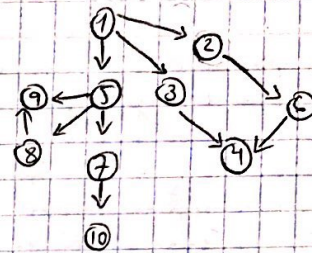
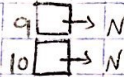
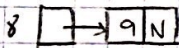
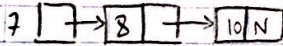
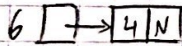
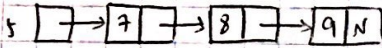
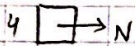
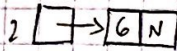
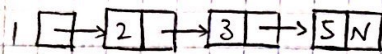
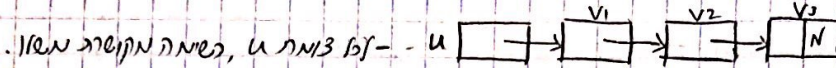


• יצירת רשת מנתונים:
 - מהני נתונים, אצור גרפים:
 - "צ" ע"י רשתות סיכום
 - "ח" ע"י רשתות סיכום

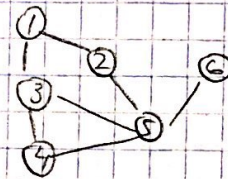
* יצירת רשתות סיכום:
 (N, E) - גרף מכוון או רשת מכוון



מסלולים - Adj[u] רשתת מקומית המכילה כל הנתונים הממוינים u. (מכאן נבנה את הרשתות)



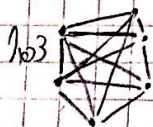
רשת מכוון:



$O(|V| + |E|)$ אם $|V| = n$ מסתמיות הוואנס הקשתות ביצור מה $M = |E|$ וכן, אצל ביצור הוואנס $O(n^2)$

אצל ביצור: אם הקשתות $|E| = O(n^2)$

אצל ביצור: אם הקשתות $|E| = O(n^2)$



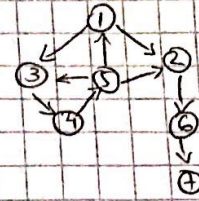
* ייצוג של מטריצת סימטיות:
מטריצת סימטיות קבועה

$|V| \times |V|$
ממ

$a_{ij} = \begin{cases} 1, & (i, j) \in E \\ 0, & \text{אחרת} \end{cases}$, $G=(V, E)$ מטריצת אדג'ן

$1 \leq i, j \leq n$

אל ממוני:

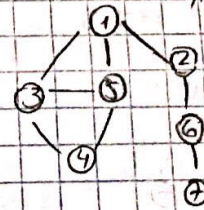


	1	2	3	4	5	6	7
1	0	1	1	0	0	0	0
2	0	1	0	0	0	1	0
3	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0
5	1	1	1	0	0	0	0
6	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0

למ גרף ב-3, ממוני שקיים מסלול קטן של 1 סימטריצה
קטנה ב-3 קיים מסלול קטן של 1 סימטריצה

המטריצה הנכונה המכילה את כל ציבורי (מסלול קטן)

אל ממוני:



	1	2	3	4	5	6	7
1	0	1	1	0	1	0	0
2	1	1	0	0	1	1	0
3	1	0	0	1	1	0	0
4	0	0	1	0	1	0	0
5	1	1	1	1	0	0	0
6	0	1	0	0	0	0	1
7	0	0	0	0	0	1	0

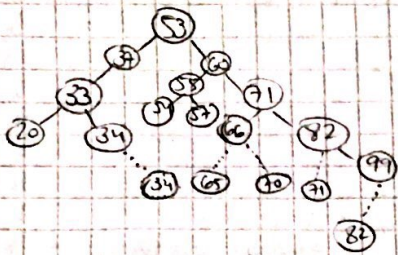
למ אולי צורה וואו גרף ב-3, למינן זמורה

אל ממוני

אפשר ואחסון רק $\frac{n^2-n}{2}$ אברים במקום n^2

$a_{ij} = a_{ji}$

למ מטריצת סימטיות



סמוכות חשש (ה) h-גינה

* בהתבוננות על מספר שונים יהיה כשר נחשב
 אותו "כסוף" כל שגיף מחזיק שבו הוא צריך
 זרעונים ותמיד הוא יגש כחלק קלה.
 יום לא יבוא זרעונים באמצע קלס.

* אם רוצים זרעונים ברגל שבו קיים,
 מתקרה שהקצת הוא פנה אבי אומץ אומץ
 קמור מן ימני
 אם אבי רוצה זרעונים ברגל שבו נמצא כקף
 כאשר נקצ זרעונים נשתי זוכה ימני ואצ
 כחם הנסו אחריו לניסאומת ברגל הסן
 קישאילסו.

סמוכות הקסד (ה) ס

* חיפוש תמונה 65:
 $65 > 53$ - מתחילים מהשורה, יותר צפון אצ ימני.
 \downarrow
 $65 > 60$ - יותר צפון, משכיבים ימני.
 \downarrow
 $65 > 71$ - יותר קטן, משכיבים שמאל.
 \downarrow
 $65 < 66$
 \downarrow
 Null-לא נמצא

* חיפוש תמונה 34:
 $34 < 53$ - מתחילים מהשורה, יותר קטן אצ שמאל.
 \downarrow
 $34 < 37$ - יותר קטן, משכיבים שמאל.
 \downarrow
 $34 > 33$ - יותר צפון, משכיבים ימני.
 \downarrow
 $34 = 34$ - נמצא.

* התבוננות על תמונה 65:
 $65 > 53$
 \downarrow
 $65 > 60$
 \downarrow
 $65 < 71$
 \downarrow
 $65 < 66$
 \downarrow
 65 קושימאילסו

* התבוננות על תמונה 70:
 $70 > 53$
 \downarrow
 $70 > 60$
 \downarrow
 $70 < 71$
 \downarrow
 $70 > 66$
 \downarrow
 70 בן ימני 66

* התבוננות על תמונה 71:
 $71 > 53$
 \downarrow
 $71 > 60$
 \downarrow
 $71 = 71$ - התבוננות על קיים והוא לא
 אה.
 \downarrow
 $71 < 82$
 \downarrow
 71 קושימאילסו 82

* התבוננות על תמונה 34:
 $34 < 53$
 \downarrow
 $34 < 37$
 \downarrow
 $34 > 33$
 \downarrow
 $34 = 34$ - התבוננות על קיים והוא פנה
 \downarrow
 34 בן ימני 34

סביבות מקסימום (h)

- * מינימום מקסימום:
- 53 - כמעט תמיד ימין
- ↓
- 60
- ↓
- 71
- ↓
- 82
- ↓
- 99 - אין היקף, יותר, מותך 99.

סביבות מציב מינימום (h)

- * מינימום מינימום:
- 33
- ↓
- 37
- ↓
- 33
- ↓
- 20 - אין היקף, מותך 20.

• לוקר של X = הקטן מוקף הקטנים מ-X (כאשר ב) הקטנים של X. בקל שלון הקוקר של 37 הוא 53
 כי קרא קרא מנהליו של X (אם מסתובב 33 שלם וצפוי).
 זמם מקסימום אין לוקר
 כל צומת שלם או כן מיני העם הקוקר מיה צפוי מתינון. אבל אחרים זמיה בתי המי של
 ליתר בן צומת שלם או כן מיני הקוקר שלם כל המינימום אם את שלם או מיניו,
 אך מיניו ואז שמאל 33 הוא
 קרא שלון הקוקר של 53 < 33 ימין ואז שמאל 33 הוא

- (צמאות הים בן מינו):
- * מינימום לוקר של 71
- 82
- ↓
- 73 - מותך 73
- * מינימום לוקר של 60
- 71
- ↓
- 66
- ↓
- 65 - מותך 65
- * מינימום לוקר של 33
- 34 - מותך 34

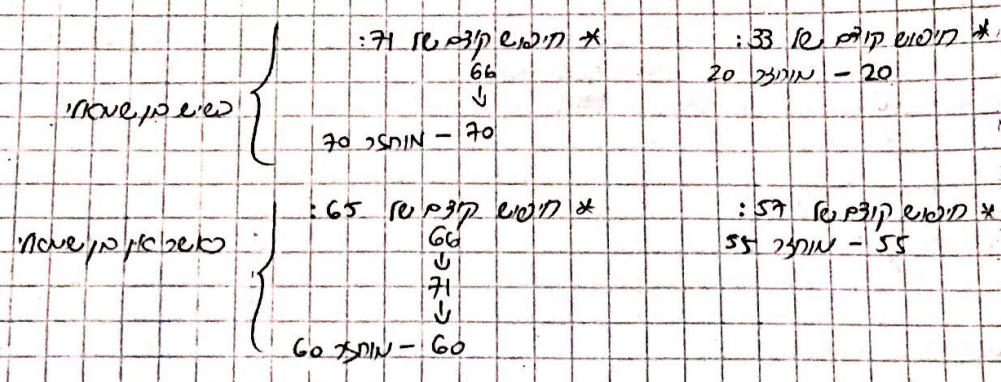
כאשר אין מינימום מסתובבם זמיה זמיה זמיה קוקר קבוצת של קבוצת של אמינו ואז אמינו

- (צמאות שלם בן מינו):
- * מינימום לוקר של 37
- 53 - מותך 53
- * מינימום לוקר של 35
- 34
- ↓
- 33
- ↓
- 37 - מותך 37
- * מינימום לוקר של 99
- 82
- ↓
- 71
- ↓
- 60
- ↓

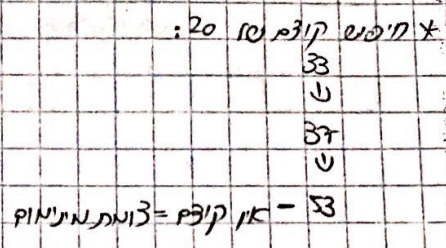
53 - אין לוקר = צומת מקסימום

סביבות מציב לוקר (h)

קופת של X = תיקון מדין הקנס X-N (כאשר הקנס שונים).
 כאשר יש קונטרול, מקום את התקופות של המן השמאל. בן שמאלה ואל ימין זך בטל.
 כאשר אין קונטרול, מוזים המה ומחשבים צורת שמוס קן ימני של אמיו, אפיו הוא הקופה.

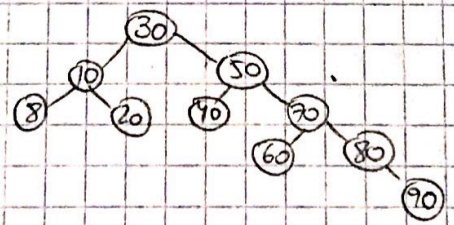


סיכויים מציאת קופה O(h)



- לוקים של הקופה זכר הצומת קצמו
 - קופה של הקופה זכר הצומת קצמו

שאלה: ד מוס קונטרול, נתונה השמת סריקה של קופה ד בסדר תמיזי.
 האם ניתן לשמור את ד מהפסימה?
 תשובה: 90, 80, 60, 70, 40, 50, 20, 8, 10, 30

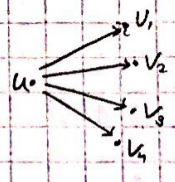


ניתן לשמור
 סריקה ד בסדר מדי עומת
 בצממון (כי חסוקים מוסים הינה).
 זמן יש סריקה תחזית וסריקה מדינת
 וזכר זכר אנשלי לשמור

השמת בסדר מדי מתימה ממיון: 90, 80, 70, 60, 50, 40, 30, 20, 10, 8

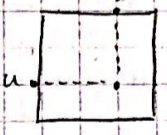
• זיכרון

* רשימות סמיכות:
 אם קוויקופ, חשמים או רשימת הסוכים. אורך הרשימה של קוויקופ u שווה למספר הסמוכים של $u = \text{מספר}$
 הקשתות היוצאות מא. $\text{Adj}(u)$
 כיוון, כל הרשימות של הקוויקופים, מכיוון אתם תסתירו של E .
 כמות הקשתות היא $|E|$ - מספר הקשתות הקלי.
 סכום אורט רשימות הסמיכות הוא $|E|$.



* מסלול סמיכות:

אם המסלול הוא $|V|^2$
 $M(u, v) = \begin{cases} 1 \\ 0 \end{cases}$



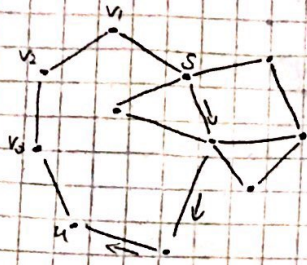
- זיכרון קטן:
 - רשימות סמיכות: $O(|E|)$
 - מסלול סמיכות: $O(|V|^2)$
 $0 \leq |E| \leq |V|^2$

הייבום למ רשימות סמיכות }
 - עבור גרף פזוז - כמות קטנה של זיכרון.
 - עבור גרף צבול - כמות גדולה של זיכרון.
 הייבום של מסלול זיכרון, כמות זיכרון תמיד גבוהה.

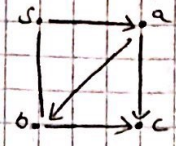
4/11/18

BFS - חיפוש רחב מתחילת

רשתות: נתון גרף $G=(V,E)$ בוחרים קובץ (מקור) S . מקור - Source



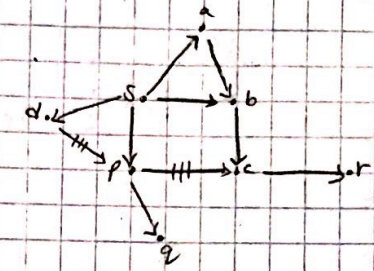
הדרכים והתקן $S \rightarrow T$ - C



$S \rightarrow a = 2$
 $S \rightarrow b = 2$
 $S \rightarrow c = 3$

כל (כאן) $\in T$ מקור

כלי כלי חיפוש רחב אינם הקובצים אלא את כל הקשתות. בעלי רגלי מניחים את הדרך הקצרה ביותר והשאר אינן נבדקות. לכן מניחים קשתות מיותרות.



$q = \pi[p]$
 $q \rightarrow p$ קובץ של p

$G=(V,E)$ גרף מכוון

* S קובץ מקור ציבורי, אסור או שמור. כל קובץ של S לא נאמן קו הוא ספין S , לאחר מכן נורה אסור וחסור שמור. קובץ של S קובצים וצד המקור.

$V[G]$ קבוצת הקובצים
 $E[G]$ קבוצת הקשתות
 $S \in V[G]$ מקור
 $u, v \in V[G]$ קובצים

```

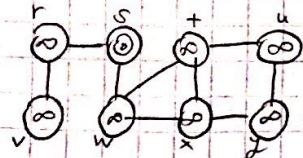
BFS(G, S) ← מציגים רשתות קבוצת השמות מוכות
for each u ∈ V[G] - S
    color[u] = WHITE
    d[u] = ∞
    π[u] = null
color[S] = GRAY
d[S] = 0
π[S] = null
Q ← S
while Q ≠ ∅
    u = head[Q]
    for each v ∈ Adj[u]
        if color[v] = WHITE
            color[v] = GRAY
            d[v] = d[u] + 1
            π[v] = u
            ENQUEUE(Q, v)
    DEQUEUE(Q)
    color[u] = BLACK
    
```

כל קובץ של G שאינו מקור
 המרחק מ- S ל- u : $d[u]$
 הקובץ של u קודם : $\pi[u]$

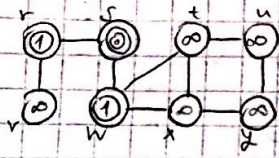
ציבורים של S הקובצים
 וקובץ וקובץ מהמרחק
 איך הוא ∞ (אינסוף)

קבוצת S מקור
 כל קובץ שמבניתי מוכח ובעולם השאר
 u קובץ של S
 כל מוקד v מוקד u , בשמור מקובץ

מרחק מ- S ל- u : $d[u]$
 קובץ של u קודם : $\pi[u]$
 v ניהוי אסור של v



Q: [s]

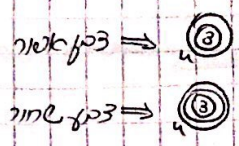


Q: [r, w]

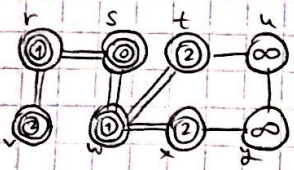


ר הוסף זיהוי שחור
 v הוסף זיהוי אפור
 U-N איומן והשיל
 קובץ s-w אוזוקים
 רסמונים ר x

רע = u .. S-N .. 3 .. 12/17 => 3
 color[r] = Gray
 d[r] = 1
 $\pi[r] = s$
 ENQUEUE(r)
 color[w] = Gray
 d[w] = 1
 $\pi[w] = r$
 ENQUEUE(w)
 DEQUEUE(s)
 color[s] = Black



w, v, t, x



Q: [t, x]
 Q: [t, x, w]
 Q: [x, w, y]

d[w] = 3

Q: [s]

Q: [s, r, w]

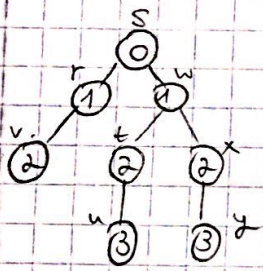
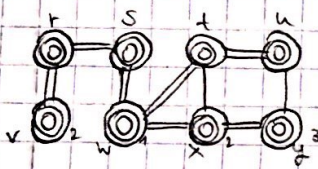
Q: [r, w]

[r, w, v]

[w, v, t, x]

[v, t, x]

[t, x]



כל הרוחב:
 (הקליטוסטיות)
 שפירוט H ב הקוב

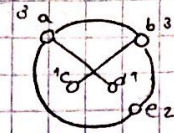
* רעורים אם הקוב שחור הקוב

$O(|V| + |E|)$ כעצ
 אם קליט קוב מ3 מ3 מ3 מ3 מ3 מ3 מ3
 מוכנה ב v אמת ב E

גרפים:

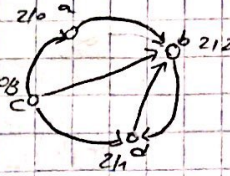
גרף - ציגור צמתים (קשתות)

מהפסימה הראשונה על אזור כמה גרפים: $(a, e), (b, e), (c, b), (a, d), (b, a)$



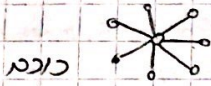
נשמע שורה ויציאות אך הם אינם הופקר. אלו גרפים לא ממונים ואין חשיבות לסדר צומת, (b, e) יכול להפוך ל- (e, b) .

כדי לקבל מכוון יש חשיבות לסדר והכתיבה כי צריך לציג את המכוון. הפסימה מתקין הנה תמיד $(a, b), (b, a), (c, b), (b, c), (c, a), (a, c)$.
 - עם (קשתות/יציאות) (לתיקון צומת).

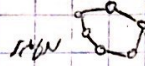


צמתים = V (ה), קשתות = E (מ).

סימני גרפים נוספים:



כוכב



מחזור



סדרון

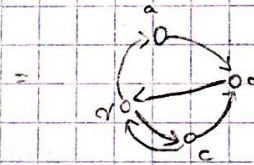
צורה של צומת: עם הקשתות שצומת קלה שיוון, (עם הקשתות שצומת קלה).
 בעזרת מכוון קיים גם צורת כעוד \rightarrow גם צורת יציאה \leftarrow
 בעזרת לא מכוון צמי 1 הוא צומת.

צ"ח גרפים:

-C- מסוימת שלבנות

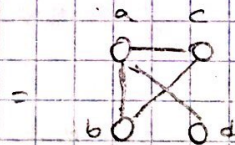
צורת המציבה שמיצקת את הוקר

	a	b	c	d
a	0	0	0	1
b	1	0	1	0
c	0	1	0	1
d	0	1	0	0

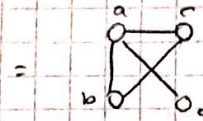
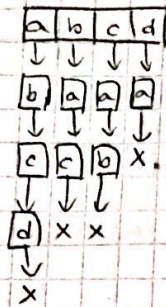
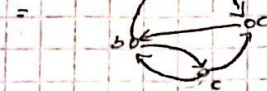
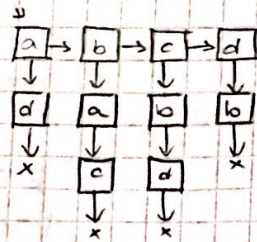


צורת המציבה שמיצקת את הוקר
 את לא מכוון יוצר מציבה סימטריה

	a	b	c	d
a	0	1	1	1
b	1	0	1	0
c	1	1	0	0
d	1	0	0	0



(אפשר רק ציור אחד)
מספר



אם היינו גוף נכון, לדוגמא עבור כל צורת את צורת הקיפוף וצורת היציאה שלו.
א- בעקבות מונח באמצעות מסלול
ב- בעקבות מונח באמצעות רשתת
תשובה:

א- $Deg_{out/IN}(A, n)$ - מטרות שכיחות של הקף, n - מספר הצמתים

מסובות: $O(n^2)/O(n^2)$

שורת 4-6 ש"ס
אחוזת מקומות
כמות $n-1$
שורת 7-12 כ"ף

הימנעו שצומבא את הצמתים -
צומבא n ב. באופן (כ"ף) זמנא כ"ס צומבא n (א"ר)

הימנעו שצומבא את הצמתים -
צומבא n ב. רשתת (כ"ף) זמנא כ"ס צומבא n (שורה)

1. Print "IN Degrees"
2. for $j \leftarrow 1$ to n do
3. $x \leftarrow 0$
4. for $i \leftarrow 1$ to n
5. $x \leftarrow x + A[i, j]$
6. print "vertex" i , "Degree" x
7. Print "out Degrees"
8. for $i \leftarrow 1$ to n do
9. $y \leftarrow 0$
10. for $j \leftarrow 1$ to n do
11. $y \leftarrow y + A[i, j]$
12. print "vertex" i , "Degree" y

א- $Deg_{IN/out}(L, n)$ - מטרות של רשתת שכיחות של הצמתים, n - מספר הצמתים

17. $p \leftarrow next(p)$
18. print "vertex" i , "Degree" y .

קמת p ש"ס מ"פ"ק של הצמתת הראשונה.
כ"ף על פ"מ"ק. לא רק כ"ף אמת ש"ס מ"פ"ק
אז תק"פ אמת $p-1$
מ"פ"ק את צמתת היציאה
מ"פ"ק את צמתת היציאה ש"ס מ"פ"ק

1. Print "out Degrees"
2. for $j \leftarrow 1$ to n do
3. $x \leftarrow 0$
4. $p \leftarrow L[i]$
5. while $(p \neq null)$ do
6. $x \leftarrow x + 1$
7. $p \leftarrow next(p)$
8. print "vertex" i , "Degree" x .
9. print "IN Degrees"
10. for $i \leftarrow 1$ to n do
11. $y \leftarrow 0$
12. for $j \leftarrow 1$ to n do
13. $p \leftarrow L[j]$
14. while $(p \neq null)$ do
15. if $key(p) = i$ then
16. $y \leftarrow next(p)$

כ"ף ש"ס מ"פ"ק $O(n+E)$
מ"פ"ק 7-18 ק"פ מ"פ"ק: $O(n(n+m))$
 $O(n(n+E))$
המ"פ"ק הראשון: $O(n(n+m))$

מ"פ"ק ש"ס מ"פ"ק אמת ש"ס מ"פ"ק

המשקל של ה מרכז הקודים.

צורת יציאה:

צומת 1, צינור 2

צומת 2, צינור 1

צומת 3, צינור 1

צומת 4, צינור 2

צורת כניסה:

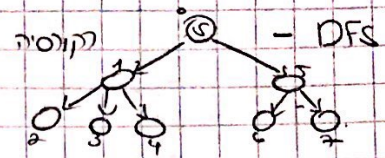
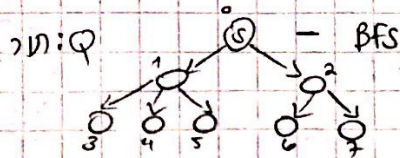
צומת 1, צינור 1

צומת 2, צינור 3

צומת 3, צינור 1

צומת 4, צינור 1

סניגור - BFS שמתחיל ב-0, סדר השתמש בתקופותיה



$d[u]$: $F[u]$! $d[u]$ אם צומת u וחסומות זמן
 detection = d
 Finish = f
 $d[u] \leq f[u]$

כל הססכים $F[u]$! $d[u]$ שאם הצמתים הם שאינם חוזרים, שונים זה מזה.
 קיים ח צמתים אז חזר חומרת זמן, נפגרו הססכים ... אז

מציבות זמן:

$Temp = Temp + 1$ (מציבת זמן מפתוחה-0)

$d[u] = Temp$ (הגיוני צומת)

$F[u] = Temp$ (סיום וקוצה קצומת)

(זמן גלגל (מכון או קלמי מכון))
 $G = (V, E)$

DFS (G)

for each $u \in V[G]$

$u \rightarrow color = White$

$\pi[u] = null$

time = 0

for each $u \in V[G]$

if ($u \rightarrow color = White$)

DFS-visit(u) - קראוהו לשיגור בקורבית

צבד צמתים, אד (שאר) צמתים
 קבוע זמן תקופות קבוע שזה זמן
 כל שזה צמתים אלא יתנו זקנים

DFS-visit(u)

$u \rightarrow color = Gray$

time = time + 1

$d[u] = time$

for each $v \in Adj[u]$

if ($v \rightarrow color = White$)

$\pi[v] = u$

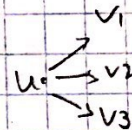
DFS-visit(v)

$u \rightarrow color = Black$

time = time + 1

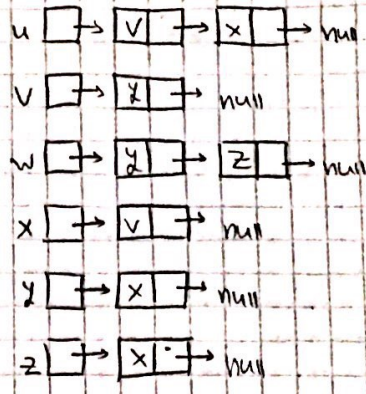
$F[u] = time$

כל צמתים v משימת הססכות
 שו u בליט d צמתים:
 זמנים שמופתחות v (X)
 אנסוף סאוס בתקופות אית
 השגרה DFS-visit

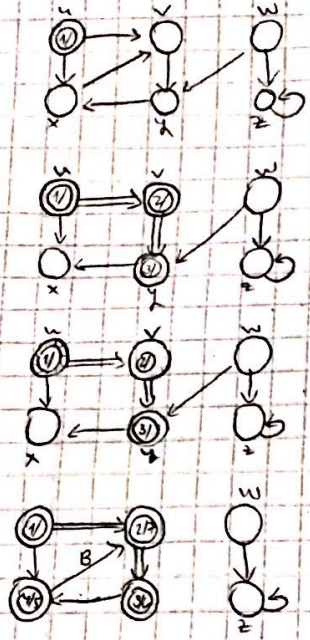


11/1/19

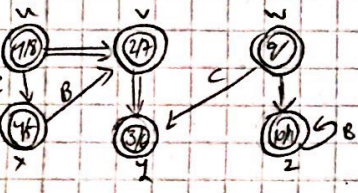
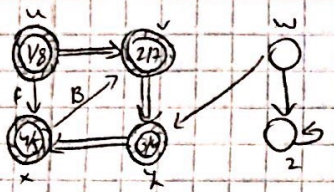
הקמת האלגוריתם לבניית:



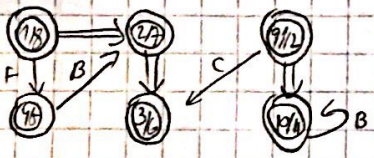
קובעים את סדר הנוצרים
הם קודם שנוצרים
ממנו



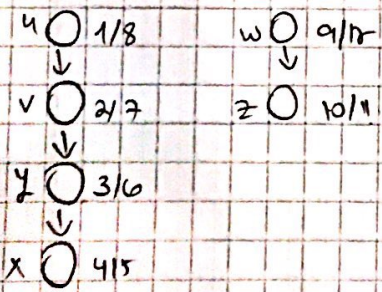
קובעים את סדר הנוצרים
הם קודם שנוצרים
ממנו
- קובעים את סדר הנוצרים
הם קודם שנוצרים
ממנו
- קובעים את סדר הנוצרים
הם קודם שנוצרים
ממנו
B = Back



לדברים אלו השתתף והתנה וכתובים ל-u ו-v!
כבר שמונים את בוקרים לעזור לה ש ל קחו
עוזר לך. הסמך את y הוא נכונ שחזר אופן
השנים שלו הסתיימו זשי ה-א וזה קוראים Cross.
n-א קוראים ל-2 (האסכולה) (y, z),
z בוצעו וכן נקבע האסור. מקרה למן סוגי
והוסף להיות קורא א (קשריפורה).



n-א אפשר לומר מן חסרה ל-z אלו לך
קרא Back, הוסף זרוע למנו מוסיפים את
השמן, ה-א מוסיפים את השמן (האחרון)



4 סוגי קשתות:
(Tree Edges) קשתות
(Back Edges) קשתות אחורה
(forward Edges) קשתות קדימה
(Cross Edges) קשתות חוצות

אם $u, v \in V$ אז $[d[u], F[u]] = [d[v], F[v]]$ - תמיד קיים שווה נקודת ריקה

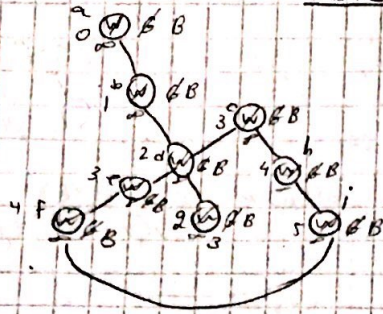
$$\alpha - [d[u], F[u]] \cap [d[v], F[v]] = \emptyset$$

$$\beta - [d[u], F[u]] \cup [d[v], F[v]] = V$$

$$\gamma - [d[u], F[u]] \supseteq [d[v], F[v]]$$

BFS

קובץ עם 10



- צבעים - דגמים
- white - w
- black - b
- Gray - G

S=a צומת מקור



* כל פעם מוזכרים את התור אחרי שהאלימנט בו ואז מנקיים את השטח שלו שנוצב או אפילו בקום. מוסיפים 1 כוס כדי לזכור שיש הקטנים שחורים ואז עוזרים נואב לזכור היציאת שחורים ואז אחרי זה נכנס לתור לזכור האלמנטים. מתחיל בקום לזכור ומסתיים בקום שחור. דגם מייצג את התהליך מצומת המקור את המקור תמיד מתחילים שר מקיפים, הונק ואסור ויש שחור אז קום בקום ניהיה שחור קום מהר.

c - הניסוי h - נכנס בתור שקיבל c

h - הניסויים h האופנים (שחורים) את מוזכרים אלוהם

* ההרצה תורה בצומת מקור

מה צומת נקרא סדר אחר של הרצה של האלמנטים. הסדר בושימה משנה.

- כל השטחים שחורים, אין מה להקדים לתור

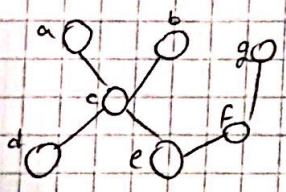
- קומה חוק נאמר האלמנטים

סיבוכיות $O(V+E)$

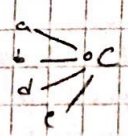
* שימוש בשטח זיכרון לזכור ולצורך את התהליך מהצומת המקור

קובץ B 333

הקובץ שניתן להפיק את צומתו ואת קצוותיו V_1, V_2 , כך שיש קשת ביניהם תוצאת צומת V_1 עם צומת V_2 . (באותה, אסור שיטת התחבר עם היציאת (הסימבול) נחלק את הקוד ככה:



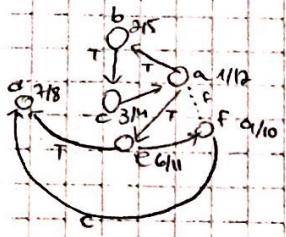
- V_1
 - a
 - b
 - d
 - e
 - g
- V_2
 - c
 - f



1) קבעתם את כל מחזורי (קשרי) פרוטוקול דאם הקול 13 333.

תשובה - מניבים BFS על הקול המקור ששמו שנפחד שיהיה. לאחר מכן נוקטים על ש הקסמה
אם יש קשת שמחברת בין דמטם עם אותו מספר - BFS - דגל אינו 13 333.
אחרת דגל ב 333.

סיבוכיות: BFS - $O(V+E)$, נמקדי על הקשתות. $O(E)$. סותרים את הקצוץ וזמי.



DFS

S=a צומת מקור =

a=1

↓

a הקשר של b = b=2

↓

משני צדדים של b = c=3

↓

אין צדדים משני צדדים של c = d=4

↓

אין צדדים משני צדדים של d = e=5

↓

הצדדים של e = f=6

↓

אין צדדים משני צדדים של f = d=7

↓

אין צדדים משני צדדים של d = c=8

↓

הצדדים של c = a=9

↓

אין צדדים משני צדדים של a = f=10

↓

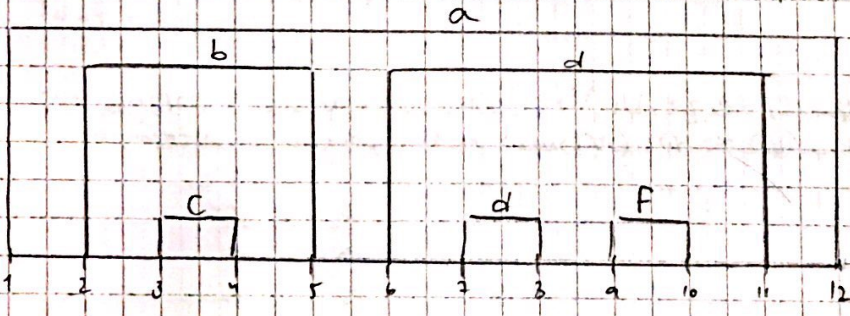
הצדדים של f = e=11

↓

הצדדים של e = a=12

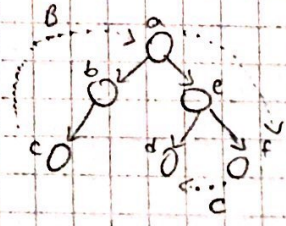
- a → 1/12
- b → 2/5
- c → 3/4
- d → 7/8
- e → 6/11
- f → 9/10

* כל צומת קצוץ 2 מספרים
 הקשר של צדדי והשם של צדדי
 * עם הצדדים הוא פי 2 ממם
 הצדדים כי נוקטים על צומת
 סמלים
 * הצומת שמתחת אינו יהיה הצומת
 שאליה הוא



ניתן לראות מהצורה תכונה של צדדים אחרת שאין סיבה לשיעור קצוץ חוטמים הם או אחרת דומה הנוני או שנים (הוכח עצמית)

- קשתות שקיימות:
- קשת של (D)
- קשתות אחרות (B) מחברות אותם הצומת מסומן הצומת שכבר היית
- קשתות קצוץ (F) אין קשר של שני. מחברות צומת קצוץ עם צומת אחרת
- אין סביון קצוץ זה הצדדים אחרת.
- קשתות חוצות (C) - אין חוצות עם הקף קצוץ/אך חוצות אחרת.
- אם היה לא אחרת מה-3 סמלים והקף של הוא חוצה.



בהינתן גרף מכוון צדדים ארוגה (כאם יש בו מעגלים).
 (מאילו נלחמת שאולי הולכת במקום מסוים בגרף של תיכור אומי (מקומות שסביר הייתי).
 (תשובה - מקובלים DFS על הגרף ובדוקים האם אזור המרכזה יש בגרף קשת אחרות,
 אם כן אז יש מקום מעגל ואם לא אין בגרף מעגל).
 (אם תכניי המקום שסביר הייתי לביאורם שחייב להיות קשת אחרות ורק זה מעגל).

סיבוכיות: $O(V+E)$ - מקרה 4 קשתות, $O(E)$ סך הכל $O(V+E)$.