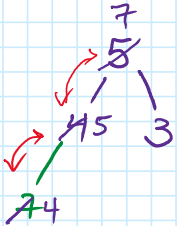
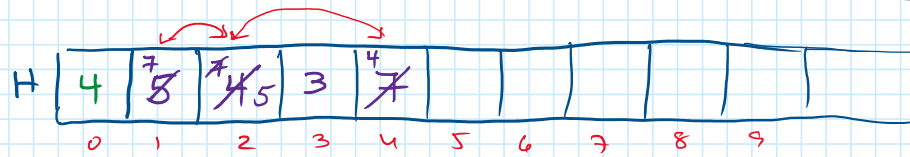
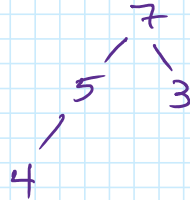


הנ"ל עדימה - הנסח מספרים אחד אחד

H++;
H[H[0]] = newnum



=>



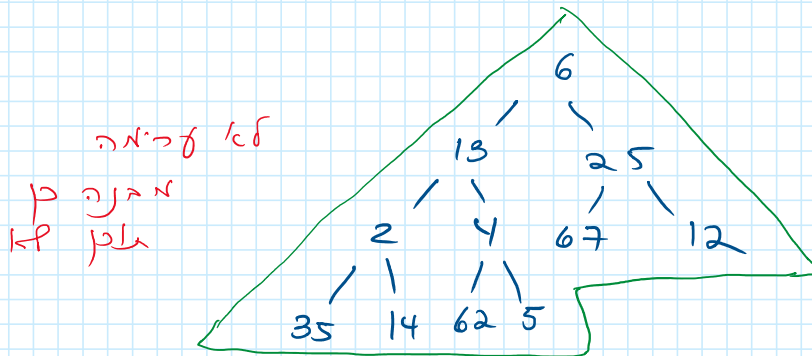
כמה זמן לוקח לבנות
עדימה של n מספרים?

$$\lg 1 + \lg 2 + \lg 3 + \dots + \lg n = O(n \lg n)$$

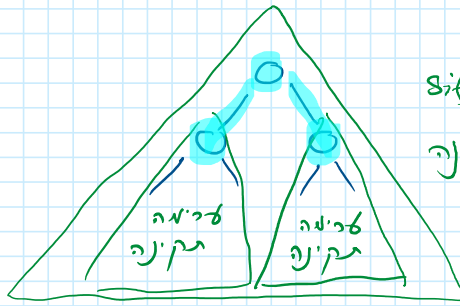
(worst case)

הנ"ל עדימה (2)

א. להבנים מספרים אחד אחד (באמצעות n...1)



באופן הנכסה, יש דרך **עדימה** אחת של n מספרים אחד אחד תבונה הילכן
פתרון - לבדוק את המסלול מהעלה לשורה
כאן יש **הרבה** דרכים של n מספרים אחד אחד תבונה הילכן



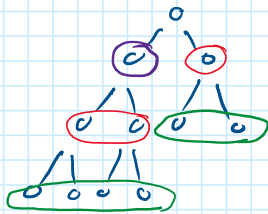
אחרי SiftDown
שנייה מקינה

נתן צוגה
שני מני הוצים הם שנייה מקינה
אולי יש בעיה * עם הצוגה והבנים

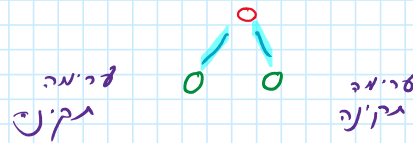
* האם קטן יותר מהבנים או לא?

אין פתרון אחר הבעיה?
Sift Down

ואם נקרא שנייה מקינה



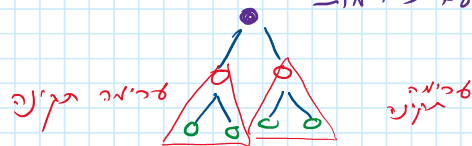
- כל העצים הם שנייה מקינה
שנייה מקינה אחת
צוגה שנייה מקינה
שנייה מקינה 2 חלק



SiftDown - אולי יותר מהצוגה אחת

- צוגה שנייה מקינה (2 חלק למחצית)
שנייה מקינה 3 חלק

SiftDown - אולי יותר מהצוגה 2 חלק



כמה החלק צריך חסר? בעצירה מקוצר מ

סוג צוגה	מספר החלק	כמה באחר יש?	מספר החלק
עצים	0	$n/2$	0
הנחה של עצים	1	$n/4$	$n/4$
סמים	2	$n/8$	$2 \cdot n/8$
...
אחר א	$k-1$	$n/2^k$	$n(k-1)/2^k$

$\lg n$

1

$\lg n$

עולה

$$\sum_{k=1}^{\lg n} \frac{n(k-1)}{2^k} = O(n)$$

Build Heap (int H[])

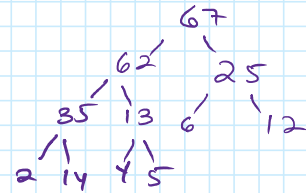
// H מערך של אינדיקסים מספיקים להבטח $n \dots 1$
// הפונקציה קונה מהמערך ערימה
{

for ($i = n/2$; $i > 0$; $i--$)

SiftDown (H, i);

}

$O(n)$



ערימה!!

H [67 62 25 35 13 6 12 2 14 4 5]

25

67 62

13 6

H [67 62 25 35 13 6 12 2 14 4 5]

0

1

2

3

4

5

6

7

8

9

10

11

↑

↑

↑

↑

↑

עזרים. קחו למקלט

אחרי ההחלפה
זה שורה של
ערימה מקינה