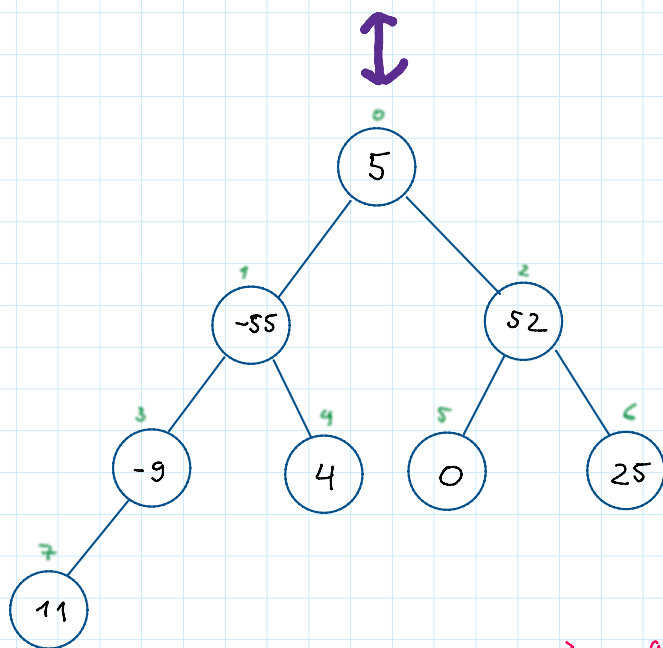


הסבר של אלגוריתם Heap Sort;

כל צרימה בינרית ניהנית להיגשה כהצדק
ול הפק:

$[5, -55, 52, -9, 4, 0, 25, 11]$



ראיון של אלגוריתם:

שלב ראשון: בינרית מהצדק במארג n נהנה
צרימה מתסומם - כך של "אבא" יהיה

לפנל מהבנים שלו, כחוצאק מהבנים -
אילו מקסימלי יהיו בלאש העצירה.

שלב שני: נחלים בין אילו ראיון לבין
אילו האחרון ואל יצי כך אילו מקסימלי
"מתם בסוף ההצדק.

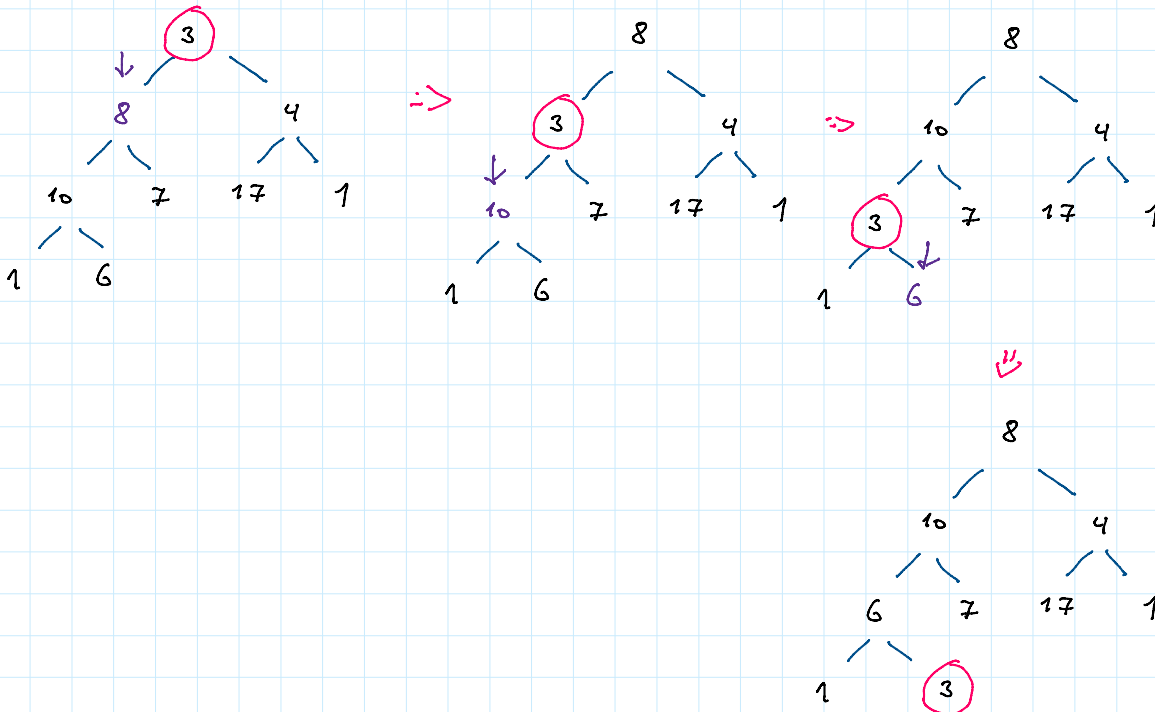
שלב שלישי: נגים באותו הצדק עלא
אילו האחרון שכבר מחוצם נכון. מחוצם זה הפלגים
1; 2 עם אותו הצדק בשול 1-n.

באופן דומה נהסין למחוצם על הפלגים
עם שלם נקבל מצדק משול 1.

איך נראה את זה?

• גרירת ערימה מקסימום בשלב 1:

נפעיל Heapify פעולה על כל איברי Heapify של הערימה. Heapify מחליפה את האב עם בנו הגדול יותר (אם יש).
 אם האב הוא 3, נפעיל Heapify על כל אחד מהבנים: 10 ו-7.
 אם האב הוא 4, נפעיל Heapify על בנו 17.



נחלק את הערימה לשתי חצויות. נפעיל את Heapify על כל אחת מהן. נחלק את הערימה לשתי חצויות. נפעיל את Heapify על כל אחת מהן. נחלק את הערימה לשתי חצויות. נפעיל את Heapify על כל אחת מהן.

אם n הוא מספר האיברים, h הוא גובה הערימה. $h = \log_2(n)$. $O(h) = O(\log n)$.
 נפעיל Heapify על כל אחד מהאיברים. $O(h)$ הוא המספר המקסימלי של פעולות Heapify שצריכות להיעשות על כל איבר.
 נפעיל Heapify על כל אחד מהאיברים. $O(h)$ הוא המספר המקסימלי של פעולות Heapify שצריכות להיעשות על כל איבר.

• **Build Max Heap** בונה ערימת מקסימום
 על ידי הפעלת **Heapify** על כל n ילדים
 מסוף ערימת המקסימום ועד יציאת כק סיבוכיות זמן
 היא $O(n \log n) = O(n \log n)$

• החלפת איבר באינדיקס זה ילד באינדיקס
 יעלה $O(1)$

• בשלבים הבאים נבנית ערימת מקסימום
 מהתבצעת הפעלת **Heapify** על אלמנט האבד בבואו.
 סך הכל יהיו $n-1$ פעמים כאלה
 וכל כן סיבוכיות זמן היא $O(n \log n)$

סך הכל סיבוכיות זמן של אלגוריתם הוא

$$O(n \log n) + O(n \log n) = O(n \log n)$$

ללא הפתרון גזי כיוון נוסף גולף וריסם

לכן סיבוכיות מקסימום היא: $O(1)$