כדי לממש את אלגוריתם דייקסטרה באופן יעיל

נצטרך לממש תור עדיפויות

Insert הכנסה ל"טבלה" S

min מציאת המינימום ב Q

delete min מחיקת המינימום מ Q ל S

Decrese key להקטין את המפתח

כל פעם ש נוסיף קודקוד ל S.

קודקוד שיש לו שכנים צריך לעדכן לכל שכן עד שלו מקסימום פעמים.

מקסימום פעמים: $n \le t \le n(n-1)$ מס' קשתות מקס'

כמה פעמים יש לעדכן כל קודקוד?

Insert n $= |V|$

min n

deletemin n

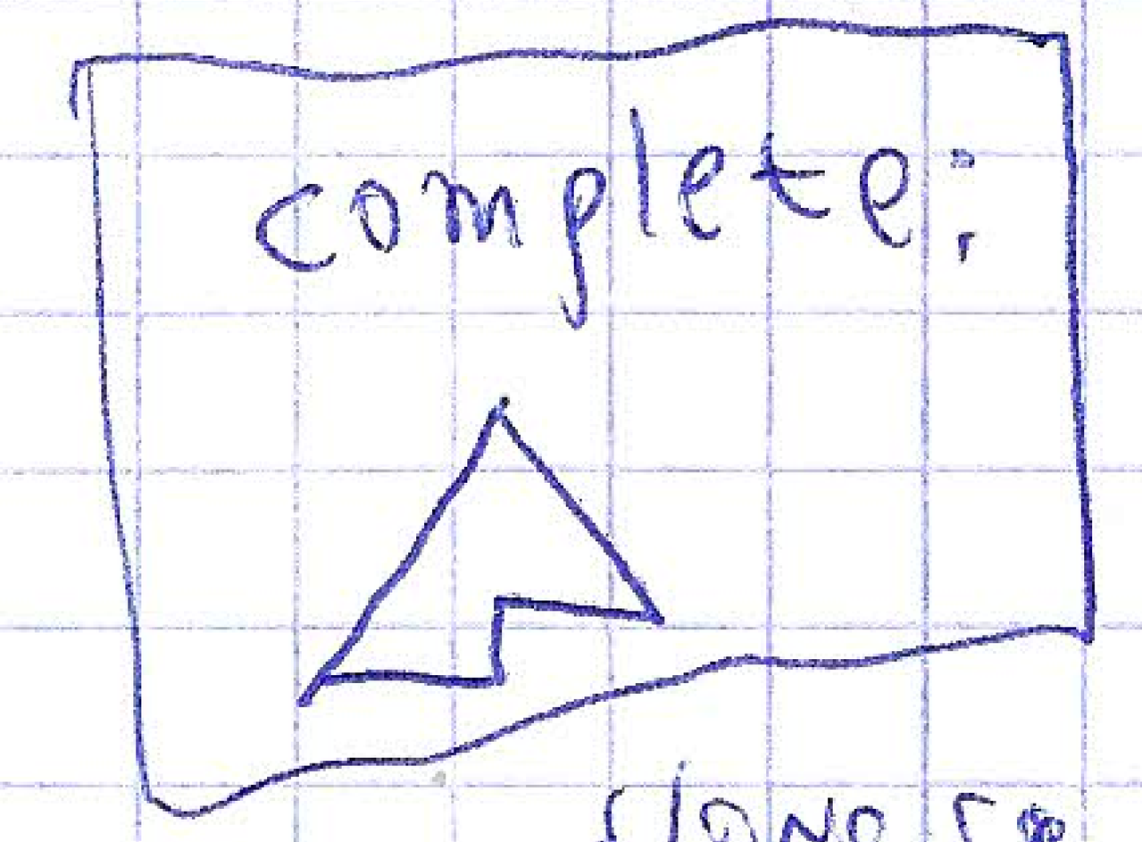
decrese key $m = |E|$

סיבוכיות בשיטה הנאיבית: $O(m \log n)$ שזה במקרה הגרוע $n^2 \log n$

ניתן לשפר עד ל $O(m + n \log n)$

שיפור זה decrese key

עצים בינאריים שלמים, כמעט מלאים, משמאל



כל קודקוד קטן משני בניו. תכונת Heap Order

הקודקוד הראשון בשורה ... בשלב ה... שמאלי

בשורש המינימום.

ניתן לשמור את העץ כמו מערך כדי לחסוך בזיכרון

Q | 7 | 23 | 17 | 24 | 29 | 26 | 19 | 65 | 33 | 40 | 79

שכבה I, שכבה II, שכבה 3, שכבה 4

Heapsort (Williams, 1964)

Insertion sort $O(n^2)$

Merge sort $O(n \log n)$

Quicksort $O(n \log n)$ expected

Heap construction $O(n)$

Heapify down $O(\log n)$

Heap extraction $O(\log n)$

Heapify down

Heap insertion $O(\log n)$

Heapify up