

מסמך נלווה – ממ"ן 13

מבנה התוכנית:

התוכנית בנויה מכמה תפריטים בהם ניתן לבחור פעולות:
תפריט ראשון – בחירה בעבודה עם רשימה ממויינת או לא ממויינת.
תפריט שני – בחירת קלט לבניית הרשימות.
תפריט שלישי – ביצוע פעולות על הרשימות הקיימות. (הדפסה, מיזוג, בנייה מחדש, והחזרת המינימום).

מבנה נתונים:

מבנה הנתונים באמצעותו ממומשת הערימה הוא רשימה מקושרת דו-כיוונית, בעלת המאפיינים הבאים:

- **Head** – מצביע לאיבר הראשון
- **Tail** – מצביע לאיבר האחרון
- **Min** – מצביע לרשומה המינימלית
- **Size** – גודל הרשימה/ערימה

בנוסף קיים משתנה סטטי בוליאני נוסף – **sorted**, באמצעותו יוחלט באיזה אופן לממש את שגרות הערימה.

שגרות התוכנית:

[makeHeap](#)

[extractMin](#)

[Union](#)

[toString](#)

makeHeap

במידה וה- Input ריק, תיווצר רשימה ריקה \ הרשימה הקיימת תיווצר מחדש.

אחרת, ייבדק הערך sorted ותרוץ השגרה המתאימה:

לרשימה ממויינת:

תרוץ השגרה createSorted

השגרה עוברת על כל האיברים שהועברו כפרמטר (כלומר n איטרציות, בהנחה שגודל המערך שהתקבל כפרמטר הוא n) וקוראת לשגרה insertSorted, אשר עוברת על הרשימה\ערימה הקיימת ומכניסה את הערך החדש במקום המתאים (כלומר k איטרציות, כאשר k גודל הערימה).

בכל איטרציה k גדל עד שמגיע לאיטרציה האחרונה – בה ערכו יהיה k+n.

לכן: createSorted תעבור n איטרציות, כאשר בכל איטרציה שלה insertSorted תעבור k עד k+n איטרציות.

כאשר insertSorted נקראת ע"י makeHeap בהכרח הערך ההתחלתי של k הוא 0.

ולכן סיבוכיות השיטה היא:

$$n * \sum_{k=1}^n k = \Theta(n^2)$$

במקרה של רשימה לא ממויינת:

השגרה תעבור על כל האיברים במערך שהועבר כפרמטר (n איטרציות) וקוראת לשגרה insertUnsorted, אשר מכניסה את האיבר לסוף הרשימה. הרשימה מחזיקה מצביע ל-Tail ולכן פעולה זו מתבצעת בזמן קבוע.

לכן במקרה של רשימה לא ממויינת, סיבוכיות השיטה היא $\Theta(n)$.

extractMin

שגרה זו מוצאת את האיבר המינימלי, מסירה אותו מהרשימה, מעדכנת את המצביע min של הרשימה לאיבר המינימלי הבא, מורידה את הערך size ב-1, ומחזירה את הערך של האיבר המינימלי שנמחק.

הרשימה שומרת תמיד מצביע לאיבר המינימלי, לכן פעולה זו לא דורשת איטרציות על הרשימה.

מכאן אופן הפעולה מתחלק;

רשימה ממויינת:

בהכרח האיבר המינימלי הבא הוא האיבר העוקב של min, ולכן min יעודכן כך שייצביע לאיבר הבא. פעולה זו מתבצעת בזמן קבוע – $O(1)$.

רשימה לא ממויינת:

לא ידוע מי האיבר המינימלי הבא ברשימה, ולכן השגרה תעבור על כל איברי הרשימה. פעולה זו תתבצע בזמן ליניארי – $\Theta(n)$.

union

שגרה זו מאחדת את הרשימות A ו-B לתוך הרשימה A.

במקרה של ערימה ממויינת:

נגדיר $m = \text{גודל הרשימה } B$, $n = \text{גודל הרשימה } A$.

השגרה תריץ m פעמים את השגרה insertSorted (אשר רצה בזמן לינארי, כגודל הרשימה A) אשר תכניס את הערך המתקבל ע"י הרצת השגרה extractMin על רשימה B (כלומר m פעמים בזמן ריצה קבוע).

מכאן שבמקרה של ערימה ממויינת סיבוכיות זמן הריצה תהיה:

$$\sum_{i=0}^{m-1} (n+i) * (m-i) = \Theta(n \lg(n))$$

במקרה של ערימה לא ממויינת:

השגרה תצביע מ-Tail של רשימה A ל-Head של רשימה B, ותאפס את המצביעים של רשימה B ותגדיר את size כ-0. פעולה זו מתבצעת בזמן קבוע – $O(1)$.

toString

ממירה את הערימה למחרוזת למטרת הדפסה – עוברת בצורה לינארית על כל איברי הערימה לפי סדרם.

סיבוכיות זמן לינארית - $\Theta(n)$