# מטלת ריצה -3

מגישים:

315539031 – גיל סמו

316013739 – אביב משה

תאור המבנה :

בחרנו לממש מבנה של ערמה יחד עם תור עדיפויות.

דרך הערמה נבצע את חיפוש המקסימום בO(1) ואת ההכנסה בO(log n) .

באמצעות תור העדיפויות נוכל למצוא את האיבר הK בגודלו ב O(klogk).

תאור הקבצים :

Heap.h – מחלקת מימוש לערמת מקסימום.

Priority\_queue.h – מחלקת מימוש של תור עדיפויות אשר מבוסס על ערמת מקסימום.

Bidder.h – מחלקת הצעה (תכונות : שם המציע וסכום).

Main.h – בניית ההגרלה עצמה ופונקציות חיפוש הזוכים.

פעולת הכנסה :

בקובץ Main קוראים לפונקציה init\_lottery שמקבלת ערמה ויוצרת 50 הצעות רנדומליות. עבור כל הצעה שיצרנו קוראים עבור הערמה לפונקציה insertNode שמכניסה את האיבר למיקום האחרון בערימה (n ) ומבצעת פעולת heapify\_up על מנת לסדר את הערמה.

מציאת מקסימום:

הפונקציה MaxBid בMain קוראת לפונקציה GetRoot של הערמה ולמעשה מחזירה את האיבר בראש הערימה.

מציאת האיבר הK :

בקובץ Main קריאה לפונקציה KBig המקבלת ערמה ומיקום K לחיפוש.

בפונקציה עצמה מכניסים את ראש הערמה אל תור העדיפויות ורצים בלולאה עד k-1 .

בתוך הלולאה מוצאים את ראש הערמה ומכניסים פנימה את שני הבנים שלו.

בשיטה זו בגלל אופן ההכנסה לתור עדיפויות תמיד נקבל את האיבר המקסימלי בראש התור.

\* אופן ההכנסה לתור העדיפויות: התור עצמו מבוסס על ערמת מקסימום לכן למעשה קוראים לפונקצית insertNode של ערמה אשר מבצעת heapify\_up .

\*הוצאת ראש התור : משתמשים בפונקציה PopRoot של ערימה אשר מחליפה את האיבר הראשון בערימה עם האחרון ומקטינה את גודל הערימה ב1. לאחר מכן מבצעים Heapify\_Down על מנת לסדר את הערימה.