פתרון לבחינה במבני נתונים ואלגוריתמים ב' – 17.1.24

שאלה 1

א.

בדומה למיון מניה, נייצג כל מחרוזת ע"י 4*26 ביטים (כל תו יכול להופיע לכל היותר 15 פעמים). לכל אחת מהמחרוזות נחשב, בזמן קבוע, את הייצוג שלה. נחפש את הערך המחושב בטבלת גיבוב. אם הוא נמצא אז מצאנו. אחרת, נוסיף אותו לטבלת הגיבוב ונמשיך לחפש.

ב.

סיבוכיות מקום – O(n) – כאשר הטבלה מתמלאת.

סיבוכיות זמן –זמן **ממוצע** O(n) .

שאלה 2

א. שתי אפשרויות לפתרון:

אפשרות 1 - להתחיל מהמינימום והמקסימום ולהתקדם בעץ בעזרת עוקב וקודם עד שנמצא פתרון או עד שנגיע לשורש. ע"י ניתוח לשיעורין אפשר להראות סיבוכיות זמן לינארית.

אפשרות 2 - בעזרת גלגולים (רוטציות) ניתן להביא את העץ למצב שבו מהשורש יוצא שרוך שמאלי ושרוך ימני, ואז הקודם ועוקב הם תמיד בפעולה אחת. כל גלגול הוא בזמן קבוע ולכן סך הגלגולים הוא לינארי.

ב.

המעבר מצומת לצומת עושה שימוש בשגרות next ו-prev , בהתאמה, שסיבוכיות הזמן שלהן היא logn. אולם, בעזרת ניתוח לשיעורין אפשר להראות כי סיבוכיות זמן הריצה היא לינארית במקרה הגרוע, עבור עץ מאוזן.

עבור עץ לא מאוזן – כאן יכול להיות יתרון בסיבוכיות הזמן. כאשר העץ לא מאוזן ויש פחות איברים, בסדרי גודל, בין שני הצדדים: במצב שאין פתרון ברגע שמגיעים מהצד "הקטן" אל השורש האלגוריתם מסתיים, וזה יכול להיות פחות מלינארי.

שאלה 3

נתחזק 2 מחסניות.

אחת עבור הכנסה ומחיקה של איברים לפי סדר 2 הפקודות הראשונות.

במחסנית השניה ישמרו ערכי המקסימום כאשר הם חדשים במבנה. כלומר, כאשר נכנס למבנה איבר מקסימלי חדש נוסיף אותו גם למחסנית השניה.

בכל ביצוע של פעולת הכנסת איבר למחסנית הראשונה, נבדוק אם יש צורך להכניסו גם למחסנית המקסימום, ובפעולת מחיקה – נבדוק אם האיבר הנמחק הוא המקסימלי אז נמחק אותו גם מתוך השניה. כך נוכל בכל שלב להחזיר את הערך המקסימלי כעת במבנה.

לגבי פעולת ההגדלה Add– נחזיק משתנה נוסף A שיהיה מאותחל ל-0. בכל פעולת הגדלה נוסיף ל-A את הערך a ונפעל בפעולת הכנסה לפי x-A ובפעולת המחיקה נוסיף את A.

בפעולת מקסימום נתייחס בהתאם – נקרא לפי התוכן + A ונכניס לפי הערך פחות A.

שאלה 4

עבור כל שורה נמצא את הערך המקסימלי, בזמן m, ונייצר ערך עבור ערמה המורכב מהערך המקסימלי וממספר השורה:

שני פתרונות אפשריים לסעיף א':

<u>אפשרות א</u>: max(i) + i/(n+1) ונמשיך לפי ערמה.

נבנה מערך עם n מקומות, ונכניס לתוכו מכל שורה את המספר המקסימלי + i/(n+1) .
נערמם לערמת מקסימום ב- O(n).
נוציא מהערמה את כמות הערכים הרצויה ב-O($\sqrt{n} * logn$).

לכן, בסעיף ב' - סיבוכיות הזמן הכוללת היא O(n), וכן סיבוכיות המקום.

<u>אפשרות ב</u>: נשאיר כזוג max(i),i ונעדכן את השגרה Heapify כך – בזמן ההשוואות (2) נשווה בין ערכי האיברים, אם הם שונים, נמשיך כרגיל. אם הם שווים, נשווה את ערכי השורות ונמשיך כרגיל.

שאלה 5

א.

האלגוריתם ימצא מספר מקסימלי רק אם כל המסלולים בין s ל- t הם זרים לגמרי (מלבד הקצוות). אחרת, ניתן לתת דוגמאות מפריכות.

ב.

I. נניח בשלילה כי המשקלים אינם שווים.
אם העפ"מ של G גדול יותר זה בסתירה להגדרתו כי הוא מכיל את G_k. אחרת, הוא קטן ממנו ומכיל קשת שלא נמצאת בו. אם הקשת הזאת היא המינימלית בחתך שהיא משרה זה בסתירה לתנאי ש- G_k קשיר, אחרת, היא לא מינימלית ויש בו קשת במשקל קטן או שווה ל-k בסתירה למינימליות בחתך.

II. נמצא עפ"מ על ידי אלגוריתם פרים או קרוסקל. תוך כדי נשמור את משקל הקשת המקסימלית בעפ"מ. הערך המוחזר הוא משקל הקשת המקסימלית בעפ"מ פחות 1.