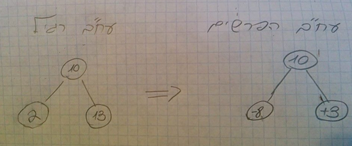
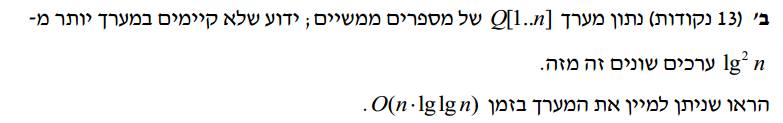


הפתרון מתבסס על עץ אדום שחור הפרשים:



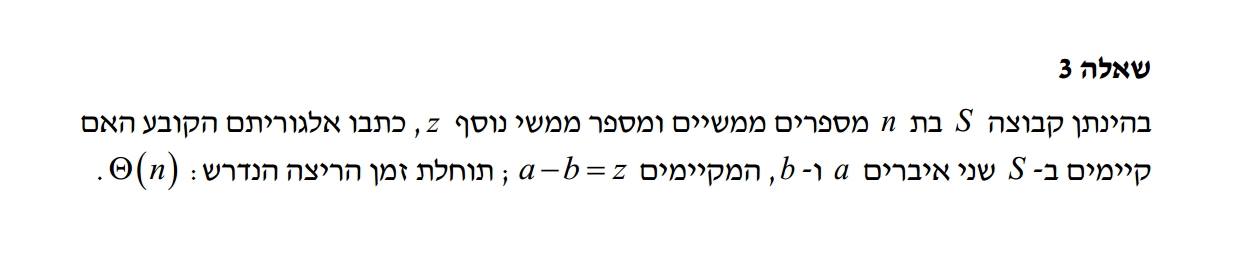
INCREASE-FROM – צריך לחפש צומת עם המפתח k (או אם לא קיים - גדול ממנו), להגדיל אותו ב-d ואם קיים בן שמאלי, להגדיל אותו גם ב-d.

כמו כן צריך לממש את SEARCH, INSERT, DELETE מחדש כך שמשנים את המפתח כל ירידה של רמה בעץ - מחשבים את ההפרש (בערך מוחלט) בין המפתח של הצומת למפתח שמחפשים / מכניסים / מוחקים.



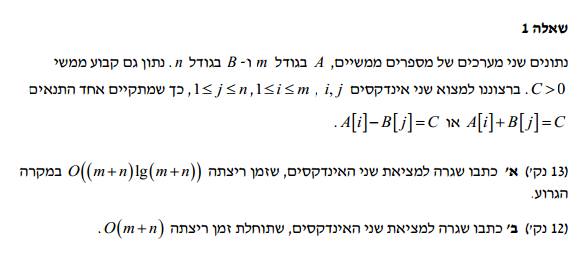
עוברים על כל איברי המערך: עבור כל אחד, אנחנו בודקים אם הוא קיים בעא"ש. אם לא - מוסיפים אותו לעא"ש.

שתי הפעולות האלו (חיפוש והוספה בעא"ש) לוקחות lgm בהנחה ויש m איברים בעא"ש, אך במקרה שלנו מספר האיברים המקסימלי הוא ואז לפי הנוסחה:



עוברים על S, לכל מספר מתייחסים אליו בתור b, מוסיפים את b + z לטבלת גיבוב (a = b + z).

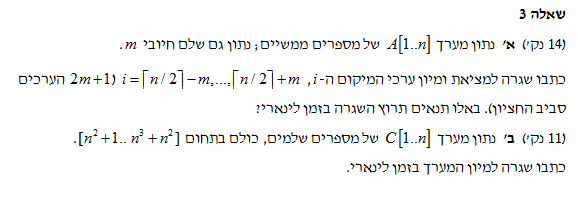
אח"כ עוברים על כל המספרים, הפעם מתייחסים אליהם בתור a ומחפשים אותם בטבלת גיבוב.



א. מיון המערך הקטן יותר. מעבר על כל האיברים במערך הלא-ממוין וחיפוש בינארי של האיבר +- C.

ב. עבור כל תא ב-B, הכנסת אינדקס לטבלת גיבוב בגודל n עם המפתח C-Bj ו C+Bj - סה"כ 2n פעולות

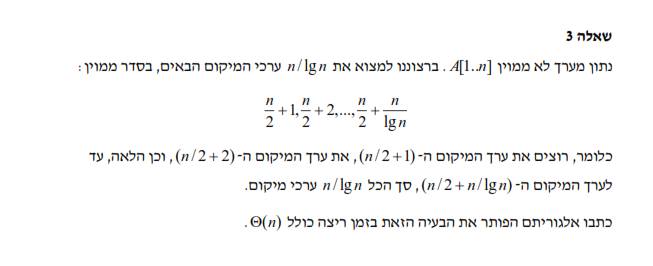
ואז בדיקה אם המפתח Ai בטבלה מוביל לתא שיש בו אינדקס - m פעולות, כל אחת בתוחלת זמן O(1).



1. מפעילים SELECT על ערכי המיקום n/2 – m ו n/2 + m כדי למצוא את ערכי הקצה. עוברים על המערך ומכניסים לתוך מערך עזר בגודל 2m+1, את כל הערכים בין min ל-max. ממיינים עם מיון-ערמה. השגרה תרוץ בזמן לינארי כל עוד mlgm=O(n).
2. עוברים על המערך, מחסרים מכל איבר ומקבלים את התחום [0..].

ניתן למיין במיון בסיס ב-O(n) לפי שאלה ז-18 במדריך הלמידה (עמ' 126).

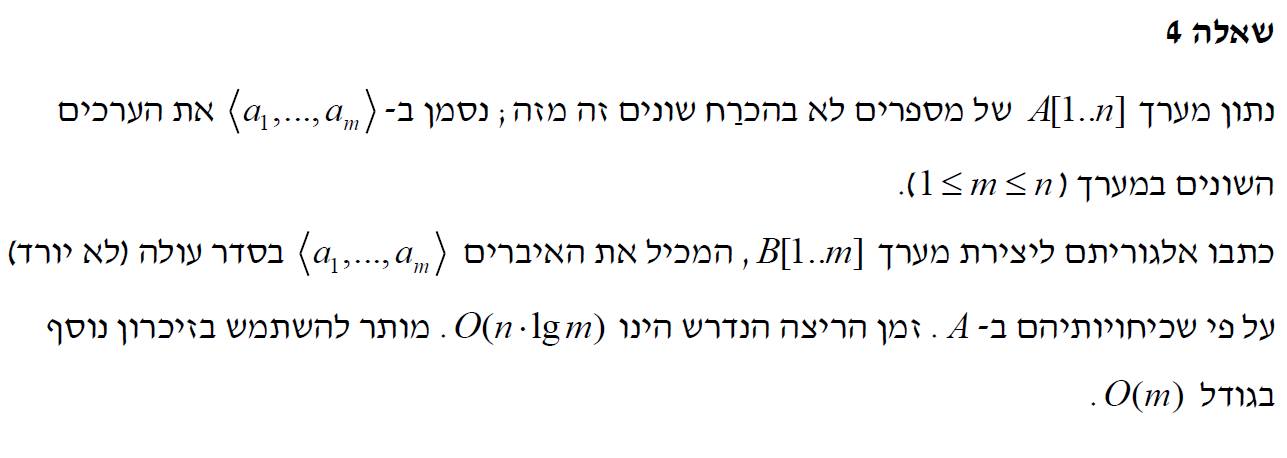
לאחר מכן מוסיפים לכל איברי המערך.



מוצאים את ערכי המיקום n/2 + 1 ו- n/2 + n/lgn בעזרת SELECT.

מכניסים לתוך מערך עזר בגודל n/lgn את כל האיברים בין שני הערכים שמצאנו.

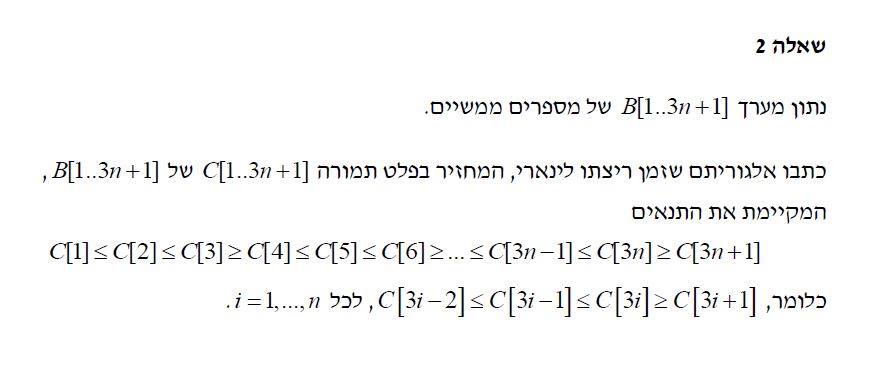
ממיינים את מערך-העזר במיון ערמה, כאשר גודלו הוא כאמור nlgn:



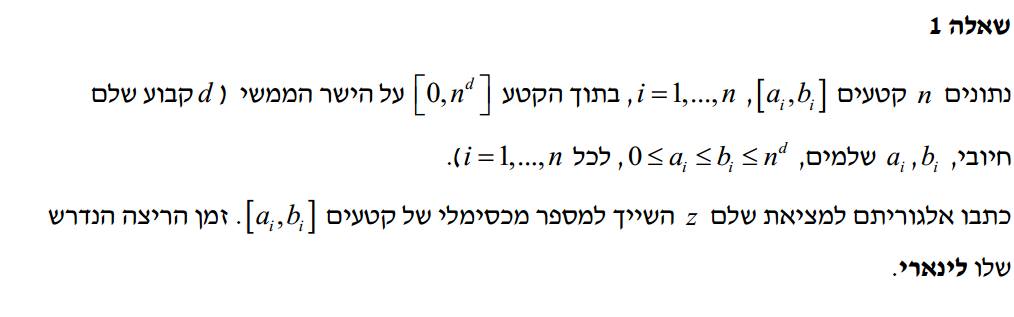
נשתמש בעת א"ש של המפתחות הייחודיים שלכל איבר נלווה שדה של מספר המופעים שלו.

בגלל שבעץ לא יותר מ-m איברים החיפוש וההכנסה אליו ב-O(lgm) אז לכל איבר במערך נחפש בעץ, אם כבר קיים נעלה את המספר מופעים ב-1 ואם לא נוסיף צומת חדש עם מספר מופעים 1.

אפשר להעתיק את הצמתים למערך בגודל m. לכל תא במערך יהיה שדה שכיחות ושדה של המפתח. נמיין בעזרת מיון מנייה על פי שדה השכיחות. השכיחות מוגבלת לתחום [1..n] ו-m<=n ולכן סיבוכיות זמן הריצה של המיון תהיה O(n). ולכן הסיבוכיות הכוללת נשארת O(n\*lgm). ניתן גם למיין לפי שכיחות בעזרת מיון-ערמה.



מתחילים בלסדר את השלשות: B[1:3], B[4:6]... יש n סדרות באורך קבוע, אז זה יקח O(n) ואחכ לעבור על הקצוות: אם b3 קטן מ b4 אז להחליף בינהם. זה לא יפגע במיון של אף אחת מהסדרות. וגם זה O(n).



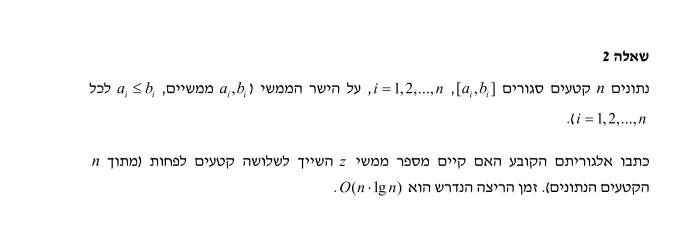
שומרים מערך של מספרים שלכל מספר יש סימון אם הוא פותח או סוגר וממלאים את כל ערכי ה-a בתור פותחים וה-b בתור סוגרים. עושים מיון בסיס לפי המפתח בבסיס n על המערך.

לפי שאלה ז-18 במדריך הלמידה (עמ' 126), העלות של כל המיון היא 2dn שזה O(n) מאחר ו-d קבוע.

עכשיו עוברים על הסדרה הממוינת ושומרים בכל רגע את מספר הקטעים הפתוחים באותו נקודה (כלומר מספר הפתיחות עד כה פחות מספר הסגירות).

בנוסף שומרים משתנה z ומשתנה שאומר מה מספר הקטעים שמכילים אותו.

כל פעם שמספר הקטעים הפתוחים בנקודה גדול מזה של z מעדכנית את z לנקודה ואת המשתנה שאומר מה מספר הקטעים הפתוחים שמכילים אותו.



למיין את הנקודות של הקטעים כאשר מוסיפים מידע נוסף של פלוס אחד לאיברי a ומינוס אחד עבור איברי b. בסוף פשוט סורקים את המערך עם מונה כאשר המספר במונה יהיה מספר הקטעים שנקודה זו שייכת אליהם.

מיזוג שני עא"ש בזמן לינארי (Merge RB Tree)

1. סריקה תוכית של כל עא"ש לתוך מערך ממוין (n).

2. מיזוג שני המערכים (n)

3. הפיכת המערך הממוין לעא"ש (n):

א. מציאת החציון ב-O(1) כי המערך ממוין והשמה שלו בשורש העץ.

ג. בתת-עץ השמאלי, קוראים לשגרה באופן רקורסיבי, כאשר האינדקסים של המערך הם של החלק השמאלי שלו (כלומר, קטנים מהחציון).

ד. בתת-עץ הימני, קוראים לשגרה באופן רקורסיבי, כאשר האינדקסים של המערך הם החלק הימני שלו.

נוסחת נסיגה: T(n) = 2T(n/2) + 1 = O(n)

ה. צובעים בשחור את כל הצמתים, ואת העלים באדום.

אפשר אחרי שתי הקריאות הרקורסיביות לבדוק - אם לצומת אין בנים (ימני ושמאלי) - אז הוא עלה וצובעים אותו באדום, אחרת שחור.

הערה: ניתן לבצע את שלבים 1 ו-3 כדי להפוך כל עח"ב לעא"ש ב-O(n).

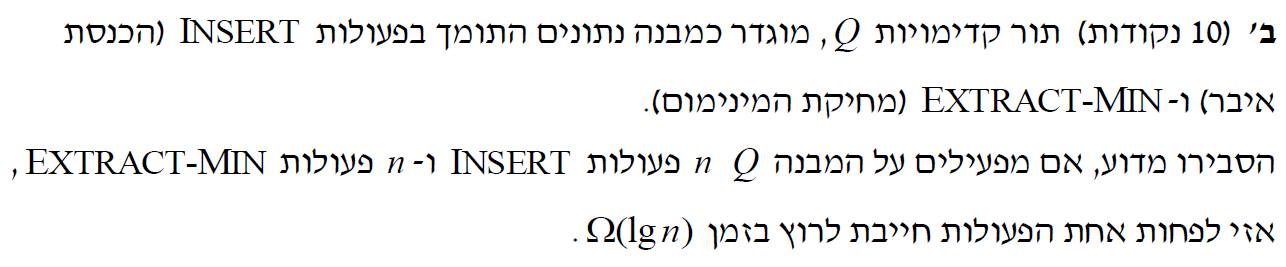
מציאת אורך מסלול הקצר/ארוך ביותר בעח"ב ב-O(n)

קצר – סריקת BFS (עם תור שלכל איבר שמכניסים שומרים מה העומק שלו) ובפעם הראשונה שמגיעים לעלה מחזירים את העומק שלו.

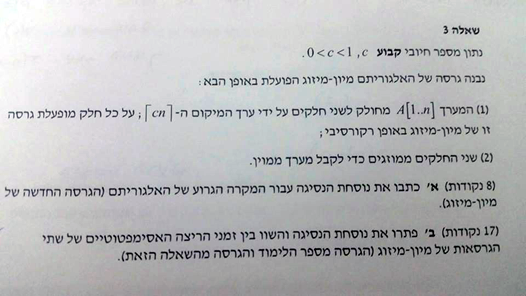
LongPath(x)

if (x == NIL or isLeaf(x)) then return 0

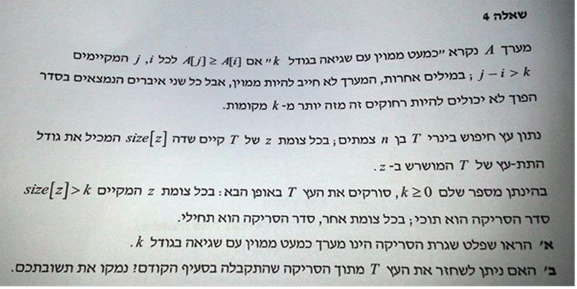
return (1 + max(LongPath(right[x]), LongPath(left[x])));



אם נניח בשלילה שכל הפעולות במקרה הגרוע אז גם במקרה הגרוע המיון יהיה זאת הסתירה. לכן נובע שלפחות אחת מהן . בגלל החסם על מיון השוואה. אם שתי הפעולות לא חסומות מלמטה אז תקבל מיון בזמן לינארי ללא שום הגבלות.



הפתרון מופיע בספר הלימוד, עמ' 60-61 (איור 4.2) ועמוד 126 (איור 7.4).



א. אם היינו סורקים את כל העץ בסדר תוכי היינו מקבלים מערך ממוין.

עכשיו אנחנו לוקחים קבוצות בגודל קטן שווה ל-k במערך הזה ומחליפים אותם בפרמוטציה שלהם (כלומר משנים את הסדר של האיברים) - כלומר כל איבר זז לכל היותר k מקומות מהמיקום שלו במערך הממוין במתאים ולכן אנו מקבלים מערך ממוין עם שגיאה בגודל k.

ב. נביא דוגמא נגדית עבור k=0 - הסריקה היא בעצם בסדר תוכי, עכשיו ניקח את העץ 1 בשורש ו-2 מימינו ואת העץ 2 בשורש ו-1 משמאלו - לשניהם כאן אותה סריקה ולכן לא ניתן לשחזר את העץ מתוצאות הסריקה בלבד.

עכשיו ל-k גדול יותר יהיו לנו חלקים במערך שאנחנו כן יכולים לשחזר לעץ המקורי שלהם (בכלל שהסדר סריקה הוא תחילי) אבל עדיין אי אפשר יהיה לשחזר את המבנה. למשל ניקח עץ על הצמתים 0, k+1 ו-1...k כאשר כל ה-k אחרונים באותו תת-עץ אז יש שתי אפשרויות:

0 בשורש, משמאלו k+1 ומימין ל-k+1 התת-עץ בגודל k.

או k+1 בשורש, משמאלו 0 ומימין ל-0 העץ בגודל k.

אבל אפשר להוכיח שלשני אלו יהיה אותו מערך סריקה