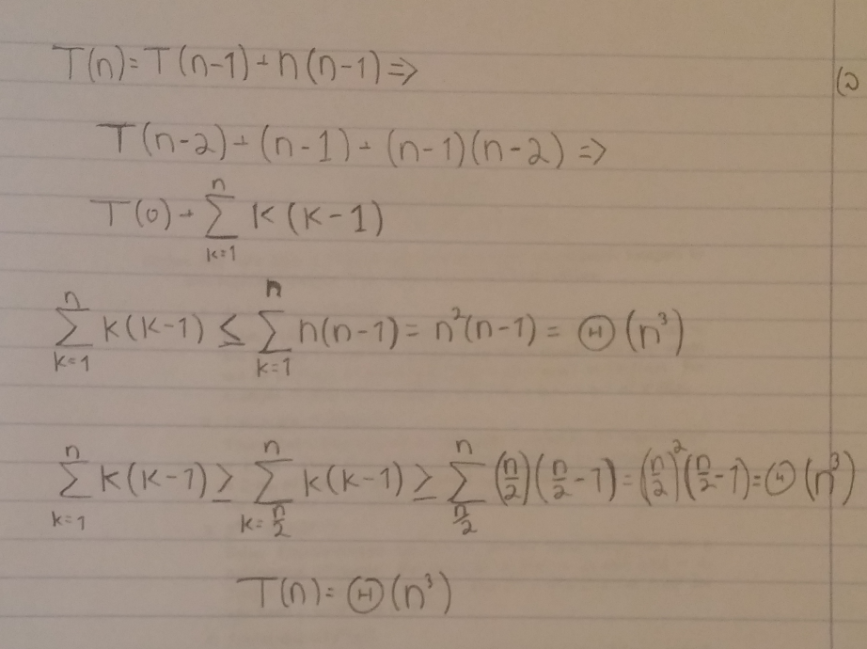
שאלה 1



שאלה 2

1. נתייחס לתאים במערך במקום מ0 עד n-1 ל1 עד n מתאמי נוחות

יהיו m=1,……,n=M אותו מערך רק בסדר ממוין, נניח בשלילה כי לכל A[I],A[j] מתקיים

A[J]-A[i]>(M-m)/n אז

M-m= (n,i=1)ΣA[J]-A[I]> (n,i=1)Σ(M-m)/n=M-m

סתירה, לכן הטענה נכונה.

CLOSE(A,I,J)

If(j-1)=1

Then return {A[i],A[j])

P🡨select (I,J,(j-i)/2)

m🡨min(A,I,J)

M🡨max(A,I,J)

If p<m+[(J-I)/2] (M-m/J-I)

Then return CLOSE (A,I,I+[(J-I)/2])

Else

Then return CLOSE (A, I+[(J-I)/2],j)

הוכחת נכונות:

עבור n=2 טריוויאלי, השגרה תחזיר כן ותחזיר את 2 התאים.

עבור n>2

I=0, J=n, p הוא האיבר במקום הn/2 ונוכיח באופן כללי עבור p, ערך מיקום k.

נניח כי

p<m+k\*[M-m/n], אז נקבל מהרקורסיה שני ערכים I ו-J כך ש 0<=A[I]-A[J]<(p-m)/n . אבל עם

הצבת p נקבל

(p-m)/k< [m+k\*((M-m)/n)-m]/k = M-m/n

ומהרקורסיה השנייה נקבל

0<=A[I]-A[j]<[(M-p)/(n-k)]<=M-m-k\*[(M-m)/n]

שאלה 3

מבנה הנתונים S, היה בנוי מעץ אדום שחור אשר נרחיב עם השדה SUM אשר היה בכל צומת ויכיל את סכום המפתחות של תת העץ.

תחזוקת שדה זה מתבצעת בזמן לוגריתמי ע"פ משפט 14.1 בספר ומכיוון שהוא תלוי רק בשני בניו, אם נבנה את העץ מלמטה למעלה כל צומת תקבל את סכום בניו ואת ערכו.

Insert: יתבצע כמו השגרה הרגיל של עץ אדום שחור (rb-insert בזמן O(lgn)), פרט לכך שבכל הכנסה נתחזק את דשה הSUM בכך שבכל ירידה בעץ נוסיף לשדה של כל צומת שעברנו את ערכו של האיבר הנכנס, זה מתבצע ב O(lgn) ע"פ משפט 14.1, השגרה תקרא בסיום ל FIXUP, ונבצע תיקון בסיבובים על מנת לשמות על תוכנות של העץ האדום שחור.

FIND: יתבצע כמו השגרה הרגיל של עץ אדום שחור,SEARCH,אם נגיע לNIL השגרה תחזירNIL סיבוכיות זמן הריצה O(lgn),כפי שמוכח בספר ומכיוון שהגובה המקסימלי של העץ הוא lgn.

RANGSUM: נשתמש בשדה ההרחבה שלנו על מנת למצוא את המפתח K, כאשר נתחיל לחפש מהשורש נעבור על כל צומת ונבדוק:

* אם המפתח בצומת הנוכחית קטן מ-X, נסכום אותו למשתנה זמני בשם countX, את מה שיש בתוך השדה SUM באותה צומת פחות הסכום שיש לבן הימני של הצומת ונתקדם ימינה.
* אם המפתח בצומת הנוכחית גדול שווה מ-X, נתקדן שמאלה מבלי להוסיף א הערך משדה ההרחבה לסכום שאנו צוברים.

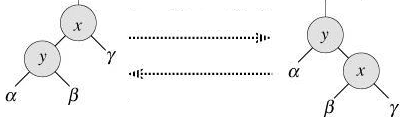
סוף חיפוש זה נשארנו עם סכום האיברים הקטנים ממש מ-X, כי סכמנו את הצמות שבו המפתח היה קטן מ-X והחסרנו את צידו הימני (המפתחות הגדולים או שווים ל-X).

באופן דומה נבצע זאת גם עבור הערך Y, נחפש בעץ ונסכום במשתנה זמני countY, את סכום האיברים הקטנים שווים לו, במידה ונתקלנו בצומת שווה ממש ל-Y, נוסיף אותה פלוס אחד לסכום השדה כדי לסכום גם את הצמות עצמו ולא רק את הקטנים מימנו.

כעת נשארנו עם 2 משתנים זמניים countY ו countX, נחשב את ההפרש בינהיים

countY – countX, ונקבל את סכום האיברים אשר קטנים שווים מY אבל גם גודלים שווים ל X,

סיבוכיות זמן הריצה: ביצענו 2 חופשים רגילים על עץ אדום שחור, כל אחד בעלות של O(lgn), החסום התחתון של חיפוש איבר בעץ הוא Ω(lgn), ולכן זמן הריצה כולל הוא Θ(lgn).

תמכיה בסיבוב:

* סכומו של Y משתה לסכומו של Y שהיה פחות הסכום של ϒ פחות 2 (צמתים X ו- ϒ עצמם).
* סכומו של X משתנה לסכומו של Y אחרי הסיבוב פלוס הסכום של ϒ פלוס 2 (בצמתים ϒ ו-Y עצמם).

שאלה 4

נעבור מלמטה למעלה על כל צומת בעץ ונרחיב אותה בשדה sum הסוכם את סכום כל כלל הערכים הנמצאים מתחתיה (ממש כמו סעיף 3 במבחן זה) ובשדה גובה d המציין באיזה גובה הצומת נמצאת.

לפי 14.1 ההרחבה בשדות אילו אינה משנה את זמן הריצה, ולכן זמן הריצה עד כה היה Θ(n).

כי עוברים בדיוק פעם אחת על כל צומת (ויש n כאלה), מספיק המידע מהאבא,צומת,בניו בשביל לבצע את העדכונים.

כעת נעבור שוב על כל צומת ונמצא את זה עם הערך sum המקסימאלי, ונחזיר את השתנה d שלו.

גם כאן זמן הריצה היה Θ(n) כי אני צריכים לעבור בדיוק על כל הצומת פעם אחת לקבלת המקסימום.

זמן הריצה הכולל הוא Θ(n)+ Θ(n)= Θ(n).

לא השתמשנו בזיכרון נוסף פרט לn האיברים שבעץ ולכן גם כאן Θ(n) + משתנה קבוע אחד אשר זוכר את המקסימום.

שאלה 5

מבנה הנתונים S היה מורכב מעץ אדום שחור מורחב בשדה שכיחויות freq, המציין לכל מפתח את כמות הפעמים שהוא מופיע במבנה, וערמת מקסימום שערכיה הם השכיחויות.

בין כל צומת של העץ היה מצביע עבור הצמות בערמת המקסימום.

init – עבור כל איבר במערך A (יש n כאלה) תריץ את השגרה insert (אשר אלה נרחיב במשך) מכיוון שזמן הריצה של insert הוא lgn))Θ ואנו מורצים פעולה זאת n פעמיים, מכיוון שיש לנו n איברים, זמן הריצה של שגירה זו היה Nlgn))Θ

insert – עבור איבר X מוספיה אותו למבנה ע"י הוספה סטנדרטית של עץ אדום שחור, אם הערך כבר קיים נוסיף 1 לשדה ה freqאליו הרחבנו את העץ. בנוסף מוספיה את הערך לערמת המקסימום, אם הערך קיים מעדכנת את הערך freq בהתאם למצביע ולערך החדש), נבצע HEAPIFY על ערמת המקסימום כדי לשמר את תכונתה, מכיוון שנכנס איבר אחד בכל פעם זמן הריצה של פעולה זו היה מרוכב מ:

הכנסה לעץ אדום שחור O(lgn)

MAX-HEAP-INSERT O(lgn), מוכח בספר.

MAX-HEAPIFY O(lgn), מוכח בספר.

O(lgn)+ O(lgn)+ O(lgn)= O(lgn)

כלומר השגרה חסומה מלמעלה ב O(lgn).

מצד שני, לא יתכן שנבצע פחות מlgn פעולות, כי אנו יכולים לעבור מסלול בגובה העץ או הערמה וגובהם לגרימתי ביחס למספר הצמתים, ולכן השגרה חסומה מלמטה ע"י Ω(lgn) .

ולכן סה"כ זמן הרצייה היה Θ(lgn).

freq – מחפשת את X בעזרת חיפוש רגיל של עץ אדום שחור ומחזירה את שדה freq אם X קיימת.

אחרת מחזירה 0.

גם כאן מוכח בספר O(lgn) כחסם עליון, ומהסבר הקודם חסומה חסומה מלמטה ע"י Ω(lgn) .

ולכן סה"כ זמן הרצייה היה Θ(lgn).

mostfreq – מחזירה את המקסימום של ערמת המקסימום ע"י heap-maximum בספר מוכח O(1) אבל חייבים לעשות לפחות פעלה 1, לכן סה"כ זמן הרצייה היה Θ(1).