פרוייקט מסכם – מטלת מנחה 16

קורס: 20407 – מבני נתונים ומבוא לאלגוריתמים

מוגש לידי - ד"ר אילן אורלוב

מגיש - אלדר וייס. ת.ז: 303169783

1. מטרת הפרוייקט : בפרוייקט זה נממש אלגוריתם פשוט עבור בעיית k האיברים הקטנים ביותר.
2. תכנון האלגוריתם :
3. תחילה, נקלוט מהמשתמש מספר k וגודל n למערך.
4. נתבקשו בתרגיל להכניס k בגודל 10,50, 100 וn בגודל 200,400,800 בהתאם.
5. תחנות הבדיקה, כפי שנתבקשנו, יהיו n1 = n/4, n2 = n/2 ו n3 =3n/4.
6. הפונקציה תממש מערך בגודל n מלא במשתנים אקראיים בטווח 0 עד 1023 כנתבקש.
7. בנוסף, ניצור מהn הנקלט תחנות הבדיקה n1,n2,n3 כנתבקש בתרגיל.
8. נבצע מימוש של עץ אדום שחור עם ערכי מיקום ונאתחל עץ ריק.
9. נכניס את המערך ללולאה בה כל איבר במערך ישמש פרמטר בפונקציית הכנסה חדשה.
10. פונקציית הכנסה זו תבדוק אם מספר הצמתים בעץ שווה לk. במידה ועדיין לא, יוכנס האיבר כערך בצומת בעץ. במידה ומספר הצמתים בעץ יגיע לכמות k, נבדוק אם האיבר הבא להזנה מהמערך קטן מהאיבר המקסימלי בעץ. במקרה זה, הפונקציה תוציא את האיבר המקסימלי מהעץ ותכניס את האיבר מהמערך.
11. במהלך לולאת ההכנסה, נבדוק אם אינדקס הלולאה שווה לאחת מתחנות הבדיקה, וכך נדע אם הכנסנו n1,n2,n3 איברים. בכל נקודה, כולל סיום קליטת כל איברי המערך, נדפיס את איברי העץ. כך למעשה נדמה אלגוריתם מקוון, כיוון שאנחו נדפיס את האיברים עד לאותו רגע.
12. בכך למעשה, נקבל את k האיברים הכי קטנים במערך, כיוון ששמרנו על כך שכמותם תהיה k ובכל פעם דאגנו להחליף כל צומת מקסימלי בעץ רק בצומת חדש שבוודאות קטן ממנו.
13. לבסוף, נדפיס את איברי העץ ונקבל את k האיברים הכי קטנים במערך.
14. זמני ריצה:
15. קליטת n משתנים לתוך מערך = O(n).
16. הכנסה – הכנסה של k איברים ראשונים תעלה O(logk) כיוון שמדובר בהכנסה טיפוסית של עץ אדום שחור.
17. חיפוש המקסימום בעץ יעלה לכל היותר O(logk) כיוון שנטייל לכל אורך העץ וגובהו logk.
18. הוצאת המקסימום תעלה O(logk) כמתואר בספר לגבי מחיקת איבר בעץ.
19. תיקון העץ יעלה (logk) בכל מחיקה של איבר מקסימלי והכנסת איבר חדש במקרה הגרוע.
20. לבסוף, נעבור על העץ ונדפיס k איברים, דבר שיעלה לנו O(k).

במקרה הכי גרוע:

1. נכניס לעץ k איברים O(logk) , נוציא את כולם ונחליפם בk איברים אחרים (O(logk)).
2. העץ תמיד יכיל k איברים, לכן ההדפסה גם במקרה הכי גרוע תדפיס רק k איברים (O(k)).

לסיכום, לאור המקרה הכי גרוע, סיבוכיות זמן הריצה של הכנסת מפתח תהיה Θ(log k).

סיבוכיות זמן הריצה של הדפסת k איברים תהיה Θ(k).

ניתוח זמני ריצה של פונקציות חשובות:

leftRotate(RedBlackNode x) – מימוש כמו בספר של סיבוב שמאלה. יעלה במקרה הגרוע O(logk).

rightRotate(RedBlackNode y) – מימוש כמו בספר של סיבוב ימינה. יעלה במקרה הגרוע O(logk).

insert(int key) – הכנסה, תעלה לנו במקרה הגרוע O(log k).

treeMaximum(RedBlackNode node) – תטייל לאורך העץ ותחזיר את הצומת המקסימלי. במקרה הכי גרוע תטייל בכל העץ ולכן זמן הריצה כגובה העץ O(logk).

remove(RedBlackNode v) - מימוש של הסרה, כמו בספר הלימוד, יעלה במקרה הגרוע O(logk).

fillRandArr(int[] arr, int limit) – ממלאת מערך בn איברים אקראיים, תעלה בכל מצב O(n).  
  
printArr(int[] arr) – תדפיס את כל המערך ותעלה גם היא O(n).

public void printKMin(int arr[], int k, int n1, int n2, int n3)  
מקבלת את המערך, את k ואת תחנות העצירה.  
תדפיס את כל איברי העץ. מכיוון שדאגנו שבעץ יהיה תמיד רק k איברים, טיול והדפסה יעלה לנו k\*O(1) + O(logk) = O(logk) + O(k) = Θ(k).

private void kInsert(int k, int element) - מקבלת את k ומשתנה ודואגת להכניסו לעץ רק במידה והוא קטן מהמפתח בצומת המקסימלי וכל עוד יש בעץ רק k איברים. כמפורט למעלה הסיבוכיות היא Θ(log k).

סיכום

לבסוף, מימשנו אלגוריתם אשר קיבל מערך בגודל n ומספר k ומדפיס את k האיברים הקטנים במערך.

מימשנו בנוסף עץ אדום שחור שהוגבל לk איברים ותומך בפעולת הכנסה בסיבוכיות Θ(log k) ובפעולת הדפסת k איברים בסיבוכיות Θ(k).

