**תרגיל בית מספר 2 – מבני נתונים**

שמות מגישים:

סער ויקטור – 312392822

אילון בן סימון - 312162951

שאלה 1

תנאי עצירה:

==

לסיכום נקבל כי:

שאלה 2

ידוע לפי משפט כי כל אלגוריתם למיון השוואות על סדרה באורך n מבצע במקרה הגרוע השוואות בין אברי הקלט. כמו כן, קיימים מקרים פרטיים שעבור תנאים מסוימים ניתן למיין מערך בזמן ריצה .

1. עבור המקרה בו המספר הגדול ביותר במערך הינו כאשר המערך מכיל n מספרים שלמים חיוביים נוכל להשתמש באלגוריתם מיון מנייה (מקיים את התנאים שלעיל) וננתח את זמן ריצת האלגוריתם: .
2. עבור המקרה בו המספר הגדול ביותר במערך הינו כאשר המערך מכיל n מספרים שלמים חיוביים נשים לב כי שלושת האלגוריתמים (מנייה, דלי ובסיס) שרצים בזמן לינארי ירוצו במקרה שלנו בסיבוכיות זמן ריצה העולה על :

מיון מנייה/דלי - כאשר האיבר ה-k (המקסימלי במערך) הינו פונקציה מעריכית התלויה ב-n סיבוכיות זמן הריצה היא .

מיון בסיס – כאשר ניקח את הבסיס להיות 3 (b=3) ונגיד כי קיימים d=n+1 ספרות סיבוכיות זמן הריצה תהיה .

לכן נשתמש באלגוריתם מיון מיזוג אשר מבצע מיון מערך בסיבוכיות זמן ריצה של .

1. עבור המקרה בו המספר הגדול ביותר במערך הינו כאשר המערך מכיל n מספרים שלמים חיוביים נעביר את כל מספרי המערך לבסיס n בזמן ריצה נשתמש באלגוריתם מיון בסיס כאשר b=n, d=16 וננתח את זמן ריצת האלגוריתם:

*שאלה 3*

*שלב א':*

נבנה ערימת מינימום בסיבוכיות זמן ריצה של .

*שלב ב':*

כעת נשתמש ב-ALG עבור מחיקת המינימום מהערימה והשמת המינימום באינדקס הראשון במערך עזר בגודל N. כמו כן כעת יש לנו ערימת מינימום חדשה בעלת N-1 איברים כאשר המינימום החדש הוא המינימום הבא בגודלו של הערימה הקודמת (לפני הפעלת ALG). סיבוכיות זמן הריצה של אלגוריתם ALG O() .

שלב ג':

נפעיל את שלב ב' n פעמים (לולאה) כאשר נכניס כל איבר מינימום לאינדקס הבא בתור במערך העזר. לבסוף נקבל מערך ממוין בסדר עולה. סה"כ סיבוכיות זמן הריצה של שלושת השלבים היא:

.

הראינו כי ניתן למיין מערך בגודל N בסיבוכיות זמן ריצה של בסתירה *למשפט שאומר שעבור מיון השוואות, זמן הריצה המינימלי הוא* - *ולכן הטענה איננה נכונה*

הוצאת המינימום

+ יצירת ערימת מינימום חדשה + השמה במערך עזר

ALG

ערימת מינימום בגודל N

O()

*שאלה 4*

*נניח בשלילה שקיים אלגוריתם למיון מערך* A *בגודל* n *בסיבוכיות זמן ריצה של* במקרה הגרוע.

ALG

מערך בגודל n המורכב מ-n/logn רצפים ממוינים

מערך ממוין בגודל n

o(nlogn)

שלב א':

עבור כל *איברים במערך נבצע מיון מיזוג .*

*ידוע שמיון מיזוג על מערך בגודל* n *לוקח ולכן במקרה שלנו סיבוכיות זמן הריצה תהיה*

*עבור כל מערך בגודל ובאופן כללי עבור*  *מערכים כלומר סיבוכיות זמן הריצה הכללית תהיה* .

שלב ב':

*הפעלת האלגוריתם* ALG -*כלומר נקבל מערך ממוין ב*- *וזו סתירה למשפט שאומר שעבור מיון השוואות, זמן הריצה המינימלי הוא* - *ולכן הטענה איננה נכונה.*

שאלה 5

נחשב את כמות העבודה בכל שורה (נחבר את כל האיברים שנמצאים באותה שורה).

שורה 1 : n.

שורה 2: .

שורה 3: .

.

.

.

שורה i: .

נמצא את אומגה ( ו-O על מנת למצוא חסם הדוק.

:

O: נניח כי וכעת נחשב את כמות העבודה במקרה הגרוע כלומר בצד בו יש חלוקה ב-ועל כן נקבל:

קיבלנו כי ולכן זמן הריצה של: שווה ל-.

שאלה 6

1.

לאחר הפעלת האלגוריתם Heap-extract max :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 9 | 50 | 21 | 14 | 7 | 60 | 23 | 25 | 30 | 10 | 66 | 75 | 33 | 77 |

הערה: באינדקס 15 תמצא הספרה 7 מפני שאלגוריתם התיקון מעתיק את האיבר האחרון במערך לאיבר הראשון (השורש בערימה).

2.

בהנתן איבר בערימה הטרינארית הנמצא באינדקס k, ניתן למצוא את שלושת בניו במקומות 3k-1,3k,3k+1 כמו כן , ניתן לראות כי בדוגמא לעיל המקומות (אינדקסים) שהצגנו אכן מקיימים את המצאות הבנים במקומות אלו.

לדוגמא: עבור k=3 (77) עבור הערימה לפני הפעולה Heap-Extract-Max שלושת בניות נמצאים באינדקיסים:

3\*3-1=8,3\*3=9,3\*3+1=10

כלומר הבנים ימצאו באינדקסים 8,9 ו-10.

3.

עבור ערימה טרינארית בעלת n איברים המקרה הגרוע יהיה ערימה שמיוצגת ע"י עץ שלם.

המחשה עבור כמות הצמתים בערימה : כאשר h מסמן את גובה הערימה.

כעת נסתכל על המקרה הגרוע כלומר כאשר העץ שלם ונמצא את גובה הערימה:

כידוע אלגוריתם heap extract max משתמש באלגוריתם heapify down ועל פיו נקבע זמן הריצה.

הפונקציה heapify down תשווה בין כל שלושת בנים (ערימה טרינארית) ולכן במקרה הגרוע תשווה שלוש פעמים עבור כל רמה בעץ ולכן סדר גודל של כמות ההשוואות בין איברי הערימה הוא:

*שאלה 7*

*שלב א':*

*נעביר את כל ה- -ים לבסיס* n *עם 5 ספרות בסיבוכיות זמן של .*

*שלב ב':*

*נשתמש באלגוריתם מיון בסיס (*Radix*) לפי ערכי (כל*  *צמוד בהתאמה ל-* שלו).

כאשר b=n ו-d=5 סיבוכיות זמן הריצה היא:

*שלב ג':*

*עבור כל נבדוק האם מתקיים במידה והתנאי מתקיים עבור אחד מן הזוגות השונים יוחזר "לא", אחרת יוחזר "כן". זמן הריצה במקרה הגרוע יהיה .*

*לסיכום:*

*זמן ריצת האלגוריתם יהיה .*

*שאלה 8:*

*נסתכל לדוגמא על המערך הבא:*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 6 | 5 | 4 | 3 | 2 | 1 | index |
| 1 | 6 | 5 | 4 | 3 | 2 | value |

ערימת מינימום של ה-value

ערימת מקסימום לפי סדר הכנסת האיברים לערימת המינימום

תאור מבני הנתונים:

* ערימת מינימום בגודל m המסודרת לפי ערכי ה-value של המערך.
* ערימת מקסימום בגודל m המסודרת לפי סדר הכנסת האיברים לערימת המינימום.
* כל איבר בערימת המקסימום והמינימום יחזיק מצביע של "התאום" שלו בערימה השנייה.

תאור הפעולות:

Init(S)-

נריץ build-heap לערימת מינימום עם מפתח .value נריץ build-heap לערימת מקסימום לפי סדר הכנסת האיברים בערימת המינימום . עבור הערימות (מינימום/מקסימום) נשמור את המצביעים הדדיים בין איברים מתאימים ואתחול משתנה ElementOrder. סיבוכיות זמן הריצה .

Pop()-

נשתמש באלגוריתם heap-extract-max על ערימת המקסימום לטובת מחיקת האיבר המקסימלי (על פי ElementOrder), סיבוכיות זמן הריצה היא . בעזרת המצביע נגיע ל"תאום" בערימת המינימום (מפתח value) ונמחק גם אותו. מחיקת התאום תתבצע באופן הבא: לוקחים את העלה האחרון (איבר אחרון במערך) שמים במקום הרלוונטי i ומורידים את גודל המערך ב-1 ולבסוף מבצעים תיקון (heapify down/up). זמן ריצת אלגוריתם של מחיקת "התאום" הוא . ולכן בסה"כ זמן הריצה יהיה = במידה ומוחקים את האיבר האחרון בערימת המקסימום/המינימום ונשארים עם ערימות ריקות נאפס את משתנה ElementOrder.

Push(k)-

תחילה נעדכן את משתנה ElementOrder כלומר נגדילו ב-1. נשתמש באלגוריתם heap-insert לטובת הכנסת האיבר לערימת המקסימום (מפתח ElementOrder). לערימת המינימום (מפתח value) נכניס ע"י אלגוריתם heap insert את האיבר הבא . נדאג לעדכן את המצביעים ההדדיים בהתאם. זמן הריצה של האלגוריתם heap insert הוא לסיכום זמן ריצת האלגוריתם יהיה:

.

Remove\_Min()-

נחזיר את ערך המפתח שבשורש ערימת המינימום ( מפתח value) – זמן הריצה .

נריץ את אלגוריתם heap-extract-min על ערימת המינימום ( מפתח value) ובעזרת המצביע נגיע ל"תאום" בערימת המקסימום ונמחק גם אותו. מחיקת התאום תתבצע באופן הבא: לוקחים את העלה האחרון (איבר אחרון במערך) שמים במקום הרלוונטי i ומורידים את גודל המערך ב-1 ולבסוף מבצעים תיקון (heapify down/up). זמן ריצת אלגוריתם של מחיקת "התאום" הוא . זמן ריצת heap-extract-min הוא . ולכן בסה"כ זמן הריצה יהיה .

Min()-

נחזיר את ערך המפתח שבשורש ערימת המינימום ( מפתח value) – זמן הריצה .