**ממן 12 – מבני נתונים ומבוא לאלגוריתמים – ברנדס איתי ת.ז.**

**שאלה 1:**

1. נפתור לפי שיטת האב מקרה 1. מתקיים:

נראה שקיים קבוע כך ש .

מתקיים לn גדול מספיק . לכן, עבור מתקיים:

ולפיכך, לפי מקרה 1 של שיטת האב, מתקיים .

1. נפתור לפי שיטת האב המורחבת מקרה 2. מתקיים:

נראה ש .

לפי פרק 3 (ע"מ 47) ראינו ש:

ועל כן:

*ולכן לפי שיטת האב המורחבת מקרה 2, מתקיים .*

1. *נפתור לפי שיטת האב מקרה 3. מתקיים:*

*נראה שקיים* קבוע כך ש .

עבור מתקיים (לפי פרק 3):

.

נבדוק את תנאי הרגולריות.

לכל מתקיים ו- .

לכן, לכל ו עבור מתקיים:

ולכן מתקיים תנאי הרגולריות. לפיכך, לפי שיטת האב מקרה 3 מתקיים:

1. *נפתור לפי שיטת האיטרציה.*

*לפי עץ רקורסיה נקבל שהאיטרציה ה-i תקח זמן ריצה של*

*הרקורסיה תסתיים כאשר נדרוש , ז"א כאשר . זאת כי מתקיים . ולכן:*

*נחשב כל טור בנפרד:*

1. *(לפי ע"מ 270 בספר)*

ולכן

נתיב במשוואה המקורית ונקבל:

(משום שהרי ).

1. ראשית נפשט ע"י הצבה והחלפת משתנים. נגדיר:

ונקבל לפי עמוד 40 במדריך הלמידה:

**נשתמש בשיטת האב מקרה 3**. מתקיים .

*נראה שקיים* קבוע כך ש .

נבחר ונקבל:

*נבדוק רגולריות. נבחר . לכל מתקיים:*

ולכן הרגולריות מתקיימת.

לפיכך לפי שיטת האב מקרה 3 נקבל ש .

*נציב כעת בחזרה ונקבל:*

*וכעת נציב בחזרה ונקבל:*

**שאלה 2:**

1. כיוון "=>" טריוויאלי – אם A הוא מערך מונז' אז לכל בחירת 2 שורות ו2 עמודות, ארבעת ההצטלבויות מקיימות את התנאי המדובר, לכן בוודאי זה יתקיים למקרים הפרטיים בו אנו בוחרים שורות ועמודות עוקבים.

כיוון "<=" נוכיח בעזרת הרמז, באינדוקציה נפרדת על השורות ועל העמודות.

נתון כי לכל ו- מתקיים:

**אינדוקציה על השורות:**

טענת האינדוקציה: לכל ו- מתקיים:

נכונות עבור נתונה.

נניח כעת נכונות עבור (\*\*) המקיים . נוכיח את הנכונות עבור :

לפי הנתון, (\*) ולכן:

(המעבר הראשון מוצדק לפי (\*), השני לפי (\*\*). )

ובכך הוכחה טענת האינדוקציה עבור כל .

**אינדוקציה על העמודות:**

טענת האינדוקציה: לכל ו- מתקיים:

נכונות עבור נתונה.

נניח כעת נכונות עבור (\*\*) המקיים . נוכיח את הנכונות עבור :

לפי הנתון, (\*) ולכן:

(המעבר הראשון מוצדק לפי (\*), השני לפי (\*\*). )

ובכך הוכחה טענת האינדוקציה עבור כל .

1. נשתמש בסעיף א' ע"מ לבחון כל "ריבוע" של ארבעה איברים סמוכים, וכך "נתקן" את המערך למערך מונז':

37 23 **25** 32

21 6 7 10

53 34 30 31

32 13 9 6

43 21 15 8

1. f(i) מוגדר להחזיר את האינדקס של העמודה השמאלית ביותר שבה נמצא איבר מינימלי של שורה i.

נניח בשלילה שקיימות שורות כך ש *. ברור ש:*

משום ש הוא האיבר המינימלי בשורה k.

ע"י חיבור המשוואות נקבל:

*ומשום שזהו מערך מונז', מתקיים לפי ההגדרה:*

*ולפיכך:*

*נבחן את 2 המקרים של :*

* 1. *🡨 , אך ולכן הוא האינדקס של העמודה השמאלית ביותר בה נמצא איבר מינימלי של שורה i, בסתירה להגדרת .*
  2. *🡨 , ולכן איננו האיבר המינימלי בשורה i, בסתירה להגדרת .*

*לכן, ההנחה שגויה, ומתקיים לכל מערך מונז' בגודל .*

1. בסעיף ג' הוכחנו שכל מערך מונז' מקיים .

בהתחשב בכך שאנו יודעים את אינדקס המינימום השמאלי של כל שורה זוגית, את אינדקס המינימום השמאלי של כל שורה אי-זוגית i נוכל לחשב תוך מקסימום

צעדים. (זאת כי מתקיים , ועל כן אפשר להתעלם מן האיברים שכן המינימום בהכרח לא מצוי בהם).

נגדיר ע"מ שנוכל לבצע פונקציית סכום פשוטה שתכלול גם את השורה הראשונה (שהיא אי-זוגית) וגם את האחרונה, במידה והיא אי-זוגית.

לפיכך, למערך מונז' A בגודל מתקיים:

(\*) בסכום השני שינינו משתנה סכימה, .

(\*\*) כי .

(\*\*\*) מציאת אינדקס המינימום השמאלי של כל שורה לוקח מקסימום n צעדים.

1. סיבוכיות ההפרדה הוא , סיבוכיות המשילה היא , וסיבוכיות איחוד התשובות הוא, כמו שהוכח בסעיף הקודם, . לכן:

( מספרים ממשיים אי-שליליים)

כעת נפתור לפי שיטת האיטרציה.

קל לראות לפי עץ רקורסיה שזמן הריצה של איטרציה i היא .

הרקורסיה תסתיים כאשר נדרוש , ז"א כאשר . זאת כי מתקיים . ולכן:

(\*) ולכן .

**שאלה 3:**

*המערך הנתון מייצג את העץ הכמעט שלם הבא:*

*45*

*32 54*

*15 28 60 65*

*18 14 48*

*נפעיל את השגרה BUILD-MIN-HEAP. הלולאה תריץ את MIN-HEAPIFY(A, i) כאשר .*

*המצב לאחר MIN-HEAPIFY(A, 5):*

*45*

*32 54*

*15 28 60 65*

*18 14 48*

***אין שינוי.***

*המצב לאחר MIN-HEAPIFY(A, 4):*

*45*

*32 54*

*14 28 60 65*

*18 15 48*

***החלפה בין*** A[4] <-> A[9]***.***

*המצב לאחר MIN-HEAPIFY(A, 3):*

*45*

*32 54*

*14 28 60 65*

*18 15 48*

***אין שינוי.***

*המצב לאחר MIN-HEAPIFY(A, 2):*

*45*

*14 54*

*32 28 60 65*

*18 15 48*

*45*

*14 54*

*15 28 60 65*

*18 32 48*

***החלפה בין*** A[2] <-> A[4] ***ואז החלפה בין*** A[4] <-> A[9]***.***

*המצב לאחר MIN-HEAPIFY(A, 1):*

*14*

*45 54*

*15 28 60 65*

*18 32 48*

*14*

*15 54*

*45 28 60 65*

*18 32 48*

*14*

*15 54*

*18 28 60 65*

*45 32 48*

***החלפה בין*** A[1] <-> A[2] ***ואז החלפה בין*** A[2] <-> A[4] ***ואז החלפה בין*** A[4] <-> A[8]***.***

**שאלה 4:**

1. ***ע"מ לייצג ערמה d-ית במערך A, נגדיר מחדש את הפונקציות LEFT, RIGHT, PARENT שהגדרנו בהגדרת ערמה.***

***משום שכרגע קיימים k בנים, ולא רק 2, במקום LEFT, RIGHT נגדיר CHILD(i, k) שיחזיר את הילד ה-kי של התא ה-iי.***

***הגדרת הפונקציות:***

1. *לכל צומת יש d בנים, ולכן גובה ערימה d-ית היא .*
2. *המימוש היעיל של EXTRACT-MAX לערימה d-ית זהה לשל ערימה בינארית (מעמוד 116 באמצע בספר הלימוד), מלבד כך שנצטרך להגדיר את MAX-HEAPIFY מחדש ע"מ שיתמוך בd בנים במקום ב2 בנים בלבד.*

*נגדיר מחדש את MAX-HEAPIFY:*

MAX-HEAPIFY(A, i)

1 largest 🡨 i

2 for k 🡨 1 to d

3 do if CHILD(i, k) > heap-size[A]

4 do break

5 if A[CHILD(i, k)] > A[largest]

6 then largest = CHILD(i, k)

7 if largest != i

8 then exchange A[i] <-> A[largest]

9 MAX-HEAPIFY(A, largest)

*הסבר האלגוריתם בקצרה:*

*הלולאה בשורות 2-6 עוברת על כל הבנים ומוצאת את אינדקס התא הגדול יותר – בינם לבין עצמם ולבין השורש.*

*בשורה 3-4 ישנה בדיקה שלא חרגנו מן המערך (כאשר לא קיימים חלק מהבנים). כאשר יש חריגה אנו יוצאים מן הלולאה.*

*שורות 7-9 מחליפות בין השורש לתא הגדול ביותר, במידה ואכן השורש איננו התא הגדול ביותר.*

*ניתוח זמן הריצה:*

*זמן הריצה של השגרה MAX-HEAPIFY שכתבנו על דומה לשגרה MAX-HEAPIFY מספר הלימוד (ע"מ 109), רק שהפעם פעולת ההשוואה בין השורש לבנים מתבצעת במקרה הגרוע ביותר d פעמים (כמספר הבנים), ולא רק פעמיים כמו המקרה הבינארי. לכן כל איטרציה של הרקורסיה רצה ב במקרה הגרוע ביותר.*

*כמו כן, במקרה הגרוע ביותר, האלגוריתם עובר על העץ מהצומת עד לעלה, ז"א עובר מספר פעמים של גובה העץ, שהוא .*

*השגרה EXTRACT-MAX מבצעת רק כמות קבועה של עבודה בנוסף לזמן הריצה הדרוש לMAX-HEAPIFY.*

*לפיכך האלגוריתם מקיים .*

1. *המימוש היעיל של INSERT לערימה d-ית זהה לשל ערימה בינארית (מעמוד 117 באמצע בספר הלימוד), מלבד כמובן ההגדרה של השגרה PARENT שהגדרנו בסעיף א'.*

*זאת משום שבאלגוריתם אנו מוסיפים צומת עם ערך זקיף לסוף המערך, ואז "מפעפעים" אותו מעלה במידת הצורך. כמובן שבאלגוריתם כאן אין לנו תלות במספר הבנים (חוץ מהנוסחא לחישוב האבא, בעזרת השגרה PARENT, כמובן).*

*זמן הריצה הוא עדיין כאורך גובה הערמה, אך הפעם גובה הערמה הוא , ולכן זמן הריצה במקרה הגרוע ביותר הוא .*

1. *נגדיר מחדש את MAX-HEAPIFY:*

INCREASE-KEY(A, i, k)

1 A[i] 🡨 max(A[i], k)

2 while i>1 and A[PARENT(i)] < A[i]

3 do exchange A[i] <-> A[PARENT(i)]

4 i 🡨PARENT(i)

*(זוהי אותה שגרה הממומשת בראשית עמוד 117, מלבד ששורות 1-3 שם הוחלפו בשורה 1 שנדרשה בשאלה שלנו).*

*הסבר האלגוריתם בקצרה:*

*האלגוריתם פועל בדיוק כמו האלגוריתם המקורי כאשר (שכן אז שורה 1 מקיימת*

*A[i] 🡨 k = max(A[i], k) ).*

*כאשר , למעשה נזרק הערך k ו נכנס ל בשורה 1 (ובכך לא קורה כלום למעשה), ואז התנאי בלולאה 2 לא מתקיים ולמעשה לא קורה דבר.*

*זאת אומרת שהאלגוריתם זהה לאלגוריתם המקורי, מלבד המצב שבו , בו במקום להעלות חריגה, האלגוריתם אינו מבצע דבר.*

*ניתוח זמן הריצה:*

*במקרה הרע ביותר, האלגוריתם רץ על כל הגובה של העץ, ובכל איטרציה מבצע כמות קבועה של עבודה (השוואה והחלפה). לכן זמן הריצה במקרה הרע ביותר היא .*