ממ"ן 11

# שאלה 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 שניה | 1 דקה | 1 שעה | 1 יום | 1 חודש | 1 שנה | 1 מאה |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  | 2801417 | 133378058 | 2755147513 | 71870856404 |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**צורת חישוב:** על פי הנתון מתבצעות פעולות בשניה, לכן לדוגמא החישוב עבור השורה הראשונה עבור שניה הוא . בכל פעם שנעלה יחידת זמן נהפוך אותה לשניות ונחשב את n. לדוגמא עבור דקה וכו'...

(קנה מידה לחישובי לשניות, דקה = 60 שניות, שעה = 60 דקות, יום = 24 שעות, חודש = 30 יום, שנה = 12 חודשים, מאה = 100 שנה).

# שאלה 2

1. כדי להוכיח את נכונות האלגוריתם נוכיח את שמורת הלולאה: בתחילת כל איטריצה של הלולאה while תת המערך A[1..i-1] לא מכיל את v.

**אתחול:** כאשר i=1 התת מערך A[1..0] הוא מערך ריק ולכן בוודאות v אין מוכל בו.

**תחזוקה:** נניח ששומרת הלולאה מתקיימת עבור i=k-1, זאת אומרת שv אין מוכל בA[1..k-1] , כעת נבדוק אם האיבר הA[k] הוא v במידה ולא מתקיים נמשיך ונבדוק את באיטרציה הבאה.

**סיום:** בהנחה ששמורת הלולאה התקיימה ואנחנו באיטרציה i=length[A]+1, ניתן להסיק כי v אין מוכל במערך A לאחר שעברנו על כל איבר בו.

1. התנאי בלולאה while הוא תנאי וגם, לכן שני התנאים ירוצו n פעמים והתנאי הראשון של הלולאה ירוץ פעם אחת נוספת, בגלל שתנאי הראשון לא יתקיים באיטרציה הn+1 לא תתבצע השוואה בתנאי השני לכן סה"כ 2n+1 פעמים.

לאחר מכן ישנה עוד השוואה ולכן סה"כ יש 2n+2 השוואות.

1. נכתוב אלגוריתם משופר לביצוע חיפוש לינארי, הרעיון הוא להוסיף לסוף המערך את האיבר אותו אנו מחפשים ובכך בעצם ניתן להיפטר מהשוואה משום שבסוף הלולאה בוודאות התנאי יתקיים. בסוף נבצע בדיקה אם i נמצא על האיבר האחרון, במידה וכן סימן שהאיבר לא נמצא במערך ונחזיר בהתאם שהאיבר לא נמצא במערך.

האלגוריתם בפסדו-קוד:

LINEAR-SEACH(A,v)

נוכיח את נכונות האלגוריתם, ההוכחה תהיה זהה לגרסא הקודמת של האלגוריתם חוץ מהסיום.

שמורת הלולאה: בתחילת כל איטריצה של הלולאה while תת המערך A[1..i-1] לא מכיל את v.

**אתחול:** כאשר i=1 התת מערך A[1..0] הוא מערך ריק ולכן בוודאות v אין מוכל בו.

**תחזוקה:** נניח ששומרת הלולאה מתקיימת עבור i=k-1, זאת אומרת שv אין מוכל בA[1..k-1] , כעת נבדוק אם האיבר הA[k] הוא v במידה ולא מתקיים נמשיך ונבדוק את באיטרציה הבאה.

**סיום:** בהנחה ששמורת הלולאה התקיימה, v לא נמצא בA[1…n-1], בגלל שהצבנו את הזקיף בסוף המערך התנאי חייב לא להתקיים ובכך בעצם לצאת מהלולאה. כעת עלינו לבדוק אם האינקס הוא בסוף המערך החדש או לפני כדי לדעת אם נמצא האיבר או לא.

מספר ההשוואות שיבצע האלגוריתם הוא n+1 פעמים בגלל שיש רק תנאי אחד בלולאת הwhile, בניגוד ל2 באלגוריתם הקודם, ועוד ההשוואה מחוץ ללולאה לכן סה"כ n+2 פעמים וזה כמחצית מכמות ההשוואות באלגוריתם המקורי.

1. נבדוק את כל המקרים האפשרים נחבר ונמצא את המקרה הממומצע, במידה והאיבר נמצא במקום הi כאשר יתבצעו i איטרציות. סך הכל אם נחבר את כל המקרים האפשריים ונחסר במספר האיברים נקבל , משוואה לינארית לכן במקרה הממוצע זמן היעילות הוא Θ(n). במקרה הגרוע ביותר, במקרה והאיבר נמצא כאיבר האחרון או בכלל לא במערך נרוץ n פעמים ולכן גם כאן זמן הריצה הוא Θ(n). בשני המקרים זמני הריצה הם Θ(n).

# שאלה 3

1. ננתח את סיבוכיות הזמן: על פי הספר עמוד 25 סיבוכיות הזמן של שגרת המיזוג על 2 מערכים ממיונים באורך m הוא (עוברים בצורה לינארית על שתי המערכים).

לאחר מכן אנו ממזגים מערך בגודל 2m עם מערך בגודל m, לפעולה הזאת בלבד סיבוכיות זמן של . נבצע את הפעולה k פעמים ונחבר את הכל ונקבל את המשוואה הבאה:

*ננתח את סיבוכיות המקום: התוצאה בסוף היא מערך ממוין אחד באורך mk. נצא מנקודת הנחה כי בתחילת האלגוריתם הגדרנו את המערך בגודל הנדרש ואנו פועלים עליו. לכן במקרה זה סיבוכיות המקום היא*

1. *אתאר אלגוריתם מיזוג יעיל יותר הפותר את הבעיה בעזרת שיטת הפרד ומשול. הרעיון הוא לחלק את k המערכים* לזוגות ולבצע בין הזוגות את המיזוג, לחזור על הפעולה שוב ושוב עד שנקבל מערך ממוין אחד באורך כנדרש. כך בעצם הורדנו את מספר פעולות המיזוג מk ל(מספר הפעמים אשר ניתן בכל פעם לפצל את k לזוגות).

ננתח את סיבוכיות הזמן: בשלב הראשון יהיו מערכים כאשר סיבוכיות זמן שיגרת המיזוג היא , *בשלב השני יהיו מערכים כאשר סיבוכיות הזמן של שגירת המיזוג שלהם תהיה . נמשיך את הפעולה עד שנבצע ישארו זוג מערכים באורך כל אחד. בסופו של דבר נקבל את הנוסחה:*

*סיבוכיות המקום של אלגוריתם זה נישאר , ניצור מערך חדש באורך הנתון ונפעל עליו.*

# שאלה 4

1. **הטענה נכונה.**

נתון לנו כי:

מהנתון אפשר להסיק כי גדול יותר מ משום שצריך לעלות את בחזקת 0 על מנת לקבל את . לכן מתקיים:

1. **הטענה לא נכונה.**

ניקח לדוגמא את הפונקציות , הפונקציות אכן מקיימות כך שלכל c קיים המקיים .

אך הביטוי לא גורר

ניקח לדוגמא

*קיבלנו סתירה. לכן הטענה לא נכונה.*

1. ***הטענה נכונה.***

*מהנתון ניתן להסיק כי (אינטואטיבית).*

*צריך להוכיח כי*

*\*מהנתון .*

1. ***הטענה לא נכונה.***

*ניקח לדוגמא את הפונקציות , ניראה כי הפונקציות אכן מקיימות כך שלכל c קיים המקיים:*

*אך לא הביטוי לא מתקיים לכל c. לדוגמא עבור נקבל סתירה:*

לכן הטענה לא נכונה.