**ממ"ן 12 אלגוריתמיקה**

**שאלה 1**

א.

נסמן ב Di את הקואורדינאטה (כאשר ),

הנוסחה המתמטית למרחק בין שתי קואורדינאטות Di ו Dj היא .

האלגוריתם

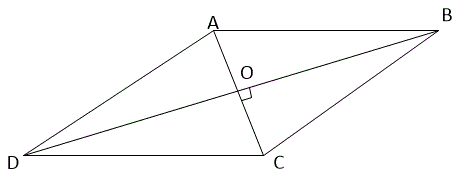
(1) נגדיר משתנה בשם min שיאותחל למרחק בין Dn לD1

(2) לכל i בין 1 ל n-1:

(2.1) אם המרחק בין Di ל Di+1 קטן מmin

(2.1.1) (המרחק בין Di ל (Di+1

הסבר – הקודקודים נתונים בכיוון השעון והמצלוע קמור לכן כל קודקוד Di סמוך ל Di+1 ו Di-1(כאשר וגם Dn וD1 סמוכים) לכן נרוץ על הקודקודים לפי הסדר הנתון ונבדוק אם מרחקם מהקודוקד הבא הוא המינינמלי, בשביל זה ניעזר במשתנה min שיאותחל למרחק בין Dn לD1 ויכיל בכל רגע נתון את המרחק המינימלי מבין המרחקים שנבדקו.

ב. לדעתי האלגוריתם שהציע הפרופסר אינו מוצלח מכיוון שהוא לא תמיד מוצא את המרחק המינימלי, קיימים מצולעים פשוטים וקמורים שהמרחק המינימלי בין שני קודקודים שלהם מתקיים עבור קודקודים שאינם סמוכים.

לדוגמא במצולע הזה:

המרחק המינימלי הוא AC והקודקודים A ו-C אינם סמוכים.

**שאלה** **2**

א.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 7 | 4 | 6 | 5 | 3 | 8 | 2 |

עבור המערך הנתון

n=8 לכן השורה הראשונה תגדיר משתנה k שערכו 8.

לאחר מכן ניכנס ללולאה (2) (הרי ), לאחר (2.1) k=8/2=4 וב(2.2) נרוץ עם i מ1עד 4k=:

תנאי (2.2.1) מתקיים רק עבור i=1,3 (

, , ,

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 3 | 4 | 2 | 5 | 7 | 8 | 6 |

) לכן נקבל את המערך:

לכן ניכנס שוב ללולאה, k=4/2=2 ונרוץ עם i מ1 עד k=2:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 3 | 4 | 2 | 5 | 6 | 8 | 7 |

לכן נבצע החלפה, לכן לא נבצע החלפה ונקבל

k לכן ניכנס שוב ללולאה, k=2/2=1 ונרוץ עם i מ1 עד k=1:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 3 | 4 | 2 | 5 | 6 | 7 | 8 |

לכן נבצע החלפה ונקבל

וכעת k=1 לכן תנאי לולאה (2) לא מתקיים ונצא ממנה.

לבסוף נחזיר את A[1]=8.

האלגוריתם בעצם מקבל מערך ומחזיר את ערך האיבר הכי גדול (נראה זאת בהוכחת נכונות בסעיף ב').

ב.

נתון שn חזקה שלמה של2, נסמן n= (כאשר b מספר שלם סופי אי שלילי).

הוכחת נכונות

בכל איטרציה מחלקים את k ב2 ובהתחלה לכן הלולאה תרוץ בדיוק b פעמים (הרי ) ואין לולאה אינסופית.

**טענה:** בכל תחילת איטרציה של לולאה (2) (כלומר כל פעם שנבדק תנאי הלולאה k>1) האינדקס של האיבר הגדול ביותר במערך קטן שווה לk, נראה זאת באינדוקציה:

בכניסה הראשונה לולאה בוודאי שהטענה מתקיימת כי הרי ולכן האינדקס של כל איברי המערך (ובפרט האיבר הגדול ביותר) קטן שווה לk.

נניח שהטענה התקיימה בריצה הx (כאשר זוהי לא האיטרציה האחרונה כלומר ) של הלולאה ונראה שמתקיימת עבור הx+1, לפי הנחת האינדוקציה האינדקס של האיבר הגדול ביותר שנסמנו בc מקיים בתחילת הריצה הx.

ב(2.1) מחלקים את k ל2 כך ש, כעת חלק ל2 מקרים אפשריים:

1. ואז לא יתקיים התנאי (2.2.1) מה שאומר שc לא ישתנה (האיבר הגדול לא יזוז) וגם בריצה הx+1 יתקיים בתחילת האיטרציה.
2. ואז בהכרח קיים בלולאה (2.2) שיקיים (כי הרי במהלך הלולאה הערכים של נעים בין ל ) מה שיגרום לביצוע החלפה בין A[i] ל A[c] וגם בריצה הx+1 יתקיים בתחילת האיטרציה.

\* אם יש כמה איברים שערכם שווה והוא הגדול ביותר במערך אז יכול להיות מצב ש

אבל אין לדבר משמעות כי אפשר להתייחס למקרה זה כמקרה הקודם (i).

לפי הטענה שזה עתה הוכחנו נקבל שבסוף הלולאה מתקיים ולכן ב(3) הערך שמוחזר הוא ערך האיבר הגדול ביותר.

זמן ריצה

לשורות (1) ו(3) יש זמן ריצה קבוע, לולאה (2) מתבצעת b פעמים כאשר ל(2.1) יש זמן ריצה קבוע ו(2.2) רצה k פעמים (הכוונה לk לאחר החילו ב2) ויש בה פקודה אחת עם זמן ריצה קבוע (2.2.1).

באיטרציה הראשונה של (2) (2.2.1) רצה פעמים ובכל איטרציה הכמות הזאת מחולקת ב2 עד 1 לכן זמן הריצה הכולל הוא ולפי הנוסחה של סכום סדרה הנדסית וקיבלנו שזמן הריצה של האלגוריתם הוא .

**שאלה** **3**

א.

לפי ההגדרה הנתונה X הוא איבר רוב ברשימה L באורך N אם ורק אם כמות המופעים של X שנסמנה בc מקיימת .

תהיי רשימה L' שמתקבלת מL על ידי מחיקת איבר אחד השווה לX ואיבר אחד השונה מX, האורך של L' הוא N'=N-2 (כי נמחקו שני איברים) וכמות המופעים של X היא c'=c-1 (נמחק בדיוק איבר אחד השווה לX) ומתקיים לכן X הוא איבר רוב גם בL'.

ב.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 4 | 4 | 3 | 1 | 2 | 3 | 3 |

נדגים את ריצת האלגוריתם על המערך

תחילה האלגוריתם מגדיר משתנה counter שמאותחל ל0 ולאחר מכן רץ בלולאה עם i מ1 עד N, נדגים את ריצת הלולאה בטבלה שמייצגת את ערך המשתנים בסוף האיטרציה:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| candidate | counter | התנאי שמתקיים | A[i] | i |
| 3 | 1 | (2.1) | 3 | 1 |
| 3 | 2 | (2.2) | 3 | 2 |
| 3 | 1 | (2.3) | 2 | 3 |
| 3 | 0 | (2.3) | 1 | 4 |
| 3 | 1 | (2.1) | 3 | 5 |
| 3 | 0 | (2.3) | 4 | 6 |
| 4 | 1 | (2.3) | 4 | 7 |

בסוף הלולאה counter=1 לכן התנאי ב(3) לא מתקיים לכן ב(4) מתבצעת קריאה לשגרה

Check-Majority-Element(A,N,4) שתחזיר False כי 4 מופיע פעמיים בA ו לכן 4 לא איבר רוב בA. האלגוריתם מחזיר תשובה שאין איבר רוב בA.

ג.

האלגוריתם כל פעם בוחר את האיבר הראשון במערך כמועמד לאיבר רוב ואז משיך לעבור על שאר האיברים עם משתנה counter שמאותחל ל1 וגדל ב1 בכל פעם שמגיעים לעוד איבר ששווה למועמד וקטן ב1 שמגיעים לאיבר שונה.

אם באיזה שהוא אינדקס j מתקיים counter=0 אז זה אומר שבתת מערך שמהאינדקס שבו נבחר המועמד עד j כמות המופעים של המועמד היא בדיוק חצי מגודל תת המערך שנבדק (הרי כל מופע של המועמד הגדיל ב1 ומופע של איבר אחר הקטין ב1) לכן האלגוריתם "ירוץ מחדש" (כלומר ייכנס שוב לתנאי (2.1) וייבחר מועמד חדש) על תת המערך שנשאר (מj+1 עד N) כי:

1. אם המועמד הוא אכן איבר רוב אז **לפי הטענה מסעיף א'** הוא יהיה איבר רוב גם בתת המערך מj+1 עד N כי לכל מופע שלו שנמחק גם נמחק איבר אחר.
2. אם יש איבר רוב ששונה מהמועמד אז כמות המופעים שלו שנמחקו קטנה או שווה לכמות המופעים של איברים שונים שנמחקו לכן הוא בהכרח יהיה איבר רוב גם בתת המערך מj+1 עד N (לפי הטענה מסעיף א' והעובדה שאם יש איבר רוב ברשימה מסוימת ומוחקים איברים ששונים ממנו הוא בוודאי יישאר איבר רוב).
3. אם אין איבר רוב בA אז או שלא יימצא איבר רוב בתת המערך מj+1 עד N ואז אין בעיה או שיימצא כזה ואז **בשורה (4)** השגרה Check-Majority-Element תחזיר False ונקבל שאין איבר רוב בA (כמו שקרה למשל במערך בסעיף ב').

ביציאה מלולאה (2) אז או שcounter=0 ואז זה אומר שגם בתת המערך האחרון שנבדק אין איבר רוב ולכן אין איבר רוב בA או ש counter>0 ואז זה אומר שנמצא איבר רוב בתת המערך האחרון שנבדק ולכן הוא מועמד להיות איבר רוב בA ולכן מתבצעת קריאה לשגרה Check-Majority-Element שתבדוק אם הוא אכן כזה.

**שאלה** **4**

א. הרעיון: כל פעם נחלק את המערך ל2 ונמצא בכל חצי את שני המספרים הגדולים ביותר ואז נמצא מבין ארבעת המספרים שנקבל את השניים הגדולים ביותר כאשר מקרי הסף בהם נצטרך לטפל הם מערכים בגודל 2 או בגודל 3.

נשתמש ב M1 בשביל לייצג את האיבר הגדול ביותר וב-M2 את השני בגודלו.

האלגוריתם יקבל אינדקסים לרוץ ביניהם מs עד f (נצא מנקודת הנחה שהקלט תקין ואכן

), נשים לב שכמות האיברים בתווך האינדקסים שנסמנה ב היא f-s+1 ובשביל לחלק את המערך.

האלגוריתם

Biggest2(L,s,f)

(1)

(2) אם

(2.1) אם אז ,

(2.2) אחרת אז ,

(3) אחרת אם

(3.1) אם אז ,

(3.2) אחרת אז ,

(3.3) אם

(3.3.1)

(3.3.2)

(3.4) אחרת אם אז

(4) אחרת

(4.1) M'1 וM'2 יקבלו את התוצאה של Biggest2(L, s,s+ )

(4.2) M''1 וM''2 יקבלו את התוצאה של Biggest2(L,, f)

(4.3) אם

(4.3.1)

(4.3.2) אם אז

(4.3.3) אחרת

(4.4) אחרת

(4.4.1)

(4.4.2) אם אז

(4.4.3) אחרת

(5) החזר את M1 וM2

בשביל לקבל את שני המספרים בגדולים ביותר בכל הרשימה נשתמש בשגרה כאשר s=1 ו- f=n.

הסבר-

בתנאים (2) ו(3) אנחנו מטפלים במקרי הסף כלומר מערכים בגודל 2 ובגודל 3 על ידי השוואות.

המקרה הכללי הוא תנאי (4) שבו אנחנו מבצעים את האלגוריתם על שני חצאי המערך בעזרת קריה רקורסיבית עם האינדקסים המתאימים כאשר נעזרים בעיגול למטה () בשביל להבטיח אינדקס שלם ואז מוצאים מהתוצאות המתקבלות את שני המספרים הגדולים ביותר על ידי השוואות.

נכונות

המספר הכי גדול במערך הוא בהכרח גם המספר הכי גדול בכל תת מערך שהוא נמצא בו ולכן ביצוע השגרה על חצי המערך שהאיבר הגדול ביותר נמצא בו (אחת מהשורות 4.1 או 4.2) תחזיר אותו בתור האיבר הגדול ביותר ואז לאחר ההשוואה בשורה (4.3) נקבל את האיבר הגדול ביותר.

האיבר השני בגודלו הוא או הגדול ביותר בחצי המערך שלו או השני בגודלו בחצי המערך שלו ובשני המקרים הביצוע הרקורסיבי של השגרה עבור אחד החצאים יחזיר אותו ואז נמצא אותו באמצעות ההשוואות.

ב.

נסמן ב את נוסחת הנסיגה,

לפי (2) עבור מערכים בגודל 2 האלגוריתם יבצע השוואה אחת לכן .

לפי (3) עבור מערכים בגודל 3 האלגוריתם יבצע עד 3 השוואות לכן

לפי (4) עבור מערכים ארוכים יותר מחלקים את המערך לשתיים ואז מבצעים את האלגוריתם על כל אחד מהתת מערכים ואז מבצעים 2 השוואות על התוצאות לכן .

כאשר n הוא חזקה שלמה של 2 השגרה תקבל רק מערכים זוגיים לכן ונקבל

ופתרון הנוסחה לפי עמוד 139 בספר ועמוד 90 במדריך הלמידה הוא

.

ג.

היא פונקציה עולה, ולפי סעיף ב' נקבל

ומצאנו חסם עליון ל.

\* אפשר גם למצוא באותה דרך חסם תחתון,

ונקבל .

**שאלה** **5**

לפי ויקיפדיה "בארוֹק הוא הסגנון האמנותי ששלט באירופה מסוף המאה ה-16 ועד אמצע המאה ה-18" לכן נניח שמלחיני הבארוק חיו בין השנים 1601 ל 1750 (גם נולדו וגם מתו בטווח השנים הזה), המספרים לא באמת חשובים לאלגוריתם אלא רק הרעיון שיש טווח סופי.

רעיון

באלגוריתם ניצור מערך שהאינדקסים שלו מייצגים את השנים בתקופת הבארוק וכל איבר בו מייצג את כמות המלחינים שהתווספו באותה שנה (מלחינים שנולדו מוסיפים 1 ומלחינים שמתו מורידים 1), מספר זה גם יכול להיות שלילי אם מתו יותר מלחינם משנולדו. לאחר יצירת המערך נעבור עליו עם משתנה שיסכום את האיברים ובכך ייצג את כמות המלחינים החיים באותה שנה וכך נמצא את השנה עם הכי הרבה מלחינים חיים.

אלגוריתם

(1) צור מערך years בגודל 150 מאותחל באפסים

(2) עבור כל מלחין, נסמן שנולד בשנת b ומת בשנת d

(2.1)

(2.2) אם אז

(3) , ,

(4) עבור y מ1 עד 150 (מייצג את השנים)

(4.1)

(4.2) אם

(4.2.1)

(4.2.2)

(5) החזר את

הסבר

תחילה ניצור את מערך השנים בשורה (1) ולאחר מכן נעבור בלולאה (2) על המלחינים ונגדיל או נקטין בהתאם את מערך השנים כך שכל שנה תייצג את כמות המלחינים שהתווספו.

נשים לב שהחסרנו 1600 מהאינדקס כי האינדקס של המערך מתחיל ב1 והשנים מתחילות ב 1601. עבור שנות המוות נחסיר שנה אחת קדימה כי אם מלחין נפטר בשנה מסוימת אז הוא עדיין שייך לקבוצת המלחינים שחיו במהלך השנה הזאת אבל לא לקבוצות החל מהשנה שאחריה, תפקיד התנאי הוא לטפל במקרה שמלחין נפטר בשנה האחרונה מה שיגרום ליציאה מגבולות המערך (בכל מקרה אנחנו לא בודקים מה קורה אחרי השנה האחרונה אז המוות לא משפיע).

בשורה (3) אנחנו מגדירים את המשתנים Max שמכיל את כמות המלחינים החיים המקסימלית בשנה מסויימת, MaxYear שמכיל את השנה שבה כמות המלחינים החיים הייתה מקסימלית וCur שמכיל את כמות המלחינים החיים הנוכחית.

בלולאה (4) רצים על השנים ומחשבים את Cur על ידי הוספת הערך הנוכחי במערך שמייצג את כמות המלחינים שהתווספו באותה שנה (נדגיש שוב, יכול להיות מספר שלילי) ואם כמות המלחינים החיים באותה שנה (Cur) גדול מכמות המלחינים החיים המקסימלית שהייתה עד עכשיו מעדכנים את Max ו MaxYear בהתאם.

בסוף בשורה (5) מחזירים את ומוסיפים 1600 (שוב, כי האינדקס של המערך מתחיל ב1).

נכונות

בעזרת הריצה על מערך השנים עם המשתנה Cur אנחנו יכולים לדעת בכל שנה כמה מלחינים היו בחיים ולפי זה למצוא את המספר המקסימלי ואת השנה הרלוונטית (על ידי השוואה בין Cur לMax).

יש לשים לב שהמשתנים Max ו MaxYear מאותחלים ב0 אבל זה בסדר כי בוודאות יש לפחות שנה אחת שבה היה לפחות מלחין אחד בחיים (מתוך ההנחה שהקלט תקין ולא ריק) ואז יתקיים התנאי (4.2) והמשתנים Max ו MaxYear יתעדכנו בהתאם.

זמן ריצה

שורות (1), (3), (5) וגם לולאה (4) רצות בזמן קבוע (אין להם תלות בגודל מערך המלחינים וטווח השנים הוא מספר קבוע).

בלולאה (2) עוברים על כל N המלחינים ובתוך הלולאה שורות (2.1) ו(2.2) רצות בזמן קבוע לכן זמן הריצה הכולל של האלגוריתם הוא .