פתרון שאלה 4 בממ"ן 11 (מיון-דירוג)

א'

אילו היינו ממיינים את המערך A באמצעות אלגוריתם מיון יציב, אזי, לפי ההגדרה, הערך $A[i]$ היה מגיע למקום $R[i]$ במערך; כל איבר של A היה מגיע למקום אחר, לכן R מהווה תמורה של $< 1,2,...,n >$.

ב'

הדרגה של $A[j]$ שווה לסכום של מספר האיברים הקטנים או שווים ל-$A[j]$ ומופיעים עד אליו (כולל הוא עצמו) ומספר האיברים הקטנים מ-$A[j]$ ומופיעים אחריו. לכל איבר $A[j]$ השגרה רצה מאינדקס 1 עד אינדקס j וסופרת את מספר האיברים הקטנים או שווים ל-$A[j]$.

איבר הקטן מ-$A[j]$ ומופיע אחריו ייספר כשהלולאה הראשית תגיע אליו.

לכן, השגרה $\mathrm{RANK}(A,R)$ מחשבת נכון את מערך הדרגות $R[1..n]$.

ג'

השגרה $\mathrm{RANK\text{-}SORT}(A)$ בונה את המערך R, המכיל באינדקס i את הדרגה של $A[i]$;

לכן, המערך U המתקבל הוא בדיוק המערך A הממוין; בסוף, מערך זה מועתק חזרה אל A.

ד'

השגרה $\mathrm{RANK}(A,R)$ מבצעת $\sum_{j=1}^{n} j = \frac{1}{2}n(n+1)$ פעולות השוואה.

האלגוריתם $\mathrm{RANK\text{-}SORT}(A)$ מבצע בנוסף $2n$ פעולות העתקה.

בסה"כ מתבצעות אפוא $\frac{1}{2}n(n+1)$ פעולות השוואה ו-$2n$ פעולות העתקה.

אם מחשיבים גם את פעולות ההעתקה במערך R, אז מתבצעות עוד $\frac{1}{2}n(n+1)$ העתקות.

אין הבדל בין מספר הפעולות במקרה הטוב ובמקרה הגרוע. האלגוריתם מבצע אותו מספר של פעולות בכל המקרים.

ה'

טענה (שמורת הלולאה): לפני האיטרציה ה-i בלולאת ה-for, $i-1$ האיברים הראשונים במערך A נמצאים כבר במקום הסופי שלהם במערך הממוין.

הוכחה:

<u>אתחול</u>: לפני הכניסה הראשונה ללולאה $i=1$ והטענה מתקיימת באופן ריק.

<u>תחזוקה</u>: נניח שהטענה נכונה לפני האיטרציה ה-i בלולאת ה-for. אם בתחילת האיטרציה ה-i מתקיים $R[i]=i$, זאת אומרת שהדרגה של $A[i]$ היא i; כלומר $A[i]$ נמצא במקום הנכון שלו. במקרה זה לולאת ה-while כלל לא מתבצעת והאיטרציה ה-i בלולאת ה-for מסתיימת מיד.

כעת מתקיים ש-i האיברים הראשונים במערך A נמצאים במקום הסופי שלהם במערך הממוין.

אם לעומת זאת מתקיים $R[i] \neq i$, אז מתחילה להתבצע לולאת ה-while. נשים לב שהדרגה של $A[i]$ לא יכולה להיות קטנה מ-i , מפני שלפי הטענה $i-1$ האיברים הראשונים במערך A נמצאים כבר במקום הסופי שלהם במערך הממוין ; כלומר, $R[i] > i$.

בכל איטרציה של לולאת ה-while מתבצעת החלפה בין $A[i]$ ל- $A[R[i]]$ (וכמובן, מתבצעת במקביל גם החלפה בין הדרגות של שני האיברים האלה במערך R). כעת האיבר $A[i]$ נמצא במקום הסופי שלו והוא לא יועבר ממנו יותר. (מדוע?) כמובן שגם האיברים במקומות $A[1..i-1]$ לא יועברו ממקומם, ומכך נובע שלולאת ה-while תסתיים לאחר $n-i$ איטרציות לכל היותר. ביציאה מלולאת ה-while מתקיים $R[i]=i$; כלומר, $A[i]$ נמצא במקום הנכון שלו.

כעת מתקיים ש-i האיברים הראשונים במערך A נמצאים במקום הסופי שלהם במערך הממוין.

<u>סיום</u> : אחרי היציאה מהלולאה מתקיים $i=n+1$ והמערך כולו ממוין.

ו'

גם באלגוריתם הזה השגרה $\text{RANK}(A,R)$ מבצעת $\frac{1}{2}n(n+1)$ השוואות.

לאחר כל החלפה $A[i] \leftrightarrow A[t]$ האיבר $A[i]$ מוצב במקום הנכון שלו, המקום $R[i]$.

לכן, לאחר $n-1$ החלפות לכל היותר המערך יהיה ממוין. האלגוריתם $\text{RANK-SORT1}(A)$ מבצע אפוא $\frac{1}{2}n(n+1)$ השוואות ועוד $3(n-1)$ העתקות במקרה הגרוע.

אם מחשיבים גם את הפעולות במערך R, אז מתבצעות בכל מקרה עוד $\frac{1}{2}n(n+1)$ העתקות (בשגרה RANK) ועוד n השוואות (בעת בדיקת תנאי הכניסה ללולאת ה-while).

כמו כן מתבצעות במקרה הגרוע עוד $n-1$ העתקות ו-$n-1$ החלפות ; כלומר, $4(n-1)$ העתקות.

ז'

דוגמה של קלט עבור המקרה הגרוע של האלגוריתם השני הינה $< n,1,2,\ldots,n-1 >$.

מערך זה משתנה באופן הבא : $< n-1,1,2,\ldots,n-2,n >$, $< n-2,1,2,\ldots,n-3,n-1,n >$, ... , $< 2,1,3,\ldots,n >$, $< 1,2,\ldots,n >$.

ח'

שני האלגוריתמים מבצעים אותו מספר של פעולות השוואה, אך במקרה הגרוע האלגוריתם השני מבצע יותר פעולות העתקה. לעומת זאת, האלגוריתם השני צורך פחות זיכרון : הוא משתמש רק במערך-עזר אחד בעוד שהאלגוריתם הראשון משתמש בשניים.

ט'

סיבוכיות הזמן של האלגוריתם השני היא $\Theta(n^2)$ גם במקרה הטוב וגם במקרה הגרוע. לכן, גם תוחלת זמן הריצה של האלגוריתם היא $\Theta(n^2)$.