סמסטר 2007א, ממ"ן 11, שאלה 1

א'

לו היינו ממיינים את המערך A בעזרת אלגוריתם 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'

כל זוג אינדקסים (i,j) המקיים $i \le j$ מופיע בלולאות פעם אחת בדיוק. המספר $R[j]$ הוא סכום מספר הזוגות (i,j) המקיימים $i \le j$, $A[i] \le A[j]$, עם מספר הזוגות (j,i) המקיימים $j > i$, $A[i] < A[j]$ (המספר השני שווה למספר הזוגות (i,j) המקיימים $i < j$, $A[i] > A[j]$).

לכן, השגרה $\mathrm{RANK}(A,R)$ פועלת נכון.

ג'

השגרה $\mathrm{RANK\text{-}SORT}(A)$ בונה את המערך R הנותן לנו את תמורת האינדקסים שעבורה המערך A ממוין; לכן, המערך U המתקבל הוא בדיוק המערך A הממוין; בסוף, מערך זה מועתק חזרה אל A.

ד'

השגרה $\mathrm{RANK}(A,R)$ מבצעת בשלב האתחול n פעולות העתקה; בתוך הלולאות מתבצעות $\sum_{j=1}^{n} j = \frac{1}{2}n(n+1)$ פעולות השוואה ואותו מספר של פעולות העתקה. לכן, $\mathrm{RANK}(A,R)$ מבצעת $\frac{1}{2}n(n+1)$ פעולות השוואה ו-$\frac{1}{2}n(n+3)$ פעולות העתקה. האלגוריתם $\mathrm{RANK\text{-}SORT}(A)$ מבצע בנוסף $3n$ פעולות העתקה, לכן סה"כ $\frac{1}{2}n(n+1)$ פעולות השוואה ו-$\frac{1}{2}n(n+9)$ פעולות העתקה.

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

ה'

נוכיח שלכל i, $1 \le i \le n$, מתקיים המשפט הבא: אם לפני הכניסה ה-i ללולאת ה-while, $i-1$ האיברים הראשונים של המערך A נמצאים כבר במקום בסדר ממוין, אזי, אחרי היציאה מהלולאה, גם האיבר ה-i נמצא במקומו בסדר ממוין.

הוכחה: נשים לב שלפני הכניסה ללולאה, $R[i] \ge i$. כל עוד $R[i] > i$, מתבצע מעבר בלולאה המעביר את $A[i]$ אל המקום $t = R[i]$ ומביא במקומו את האיבר $R[t]$ שלא היה במקום הדירוג שלו ($R[t] \ne R[i] = t$); גם אחרי ההחלפה מתקיים $R[i] \ge i$. האיברים שנמצאים במקומותיהם

(לפי הדירוג) לעולם אינם מועברים, לכן התהליך חייב להסתיים (הלולאה מסתיימת כאשר $R[i]=i$).

תנאי המשפט מתקיים באופן ריק לפני הכניסה הראשונה ללולאה ($i=1$); לכן, אחרי היציאה האחרונה מהלולאה ($i=n$), המערך כולו ממוין.

ו'

גם באלגוריתם הזה, השגרה $\text{RANK}(A,R)$ מבצעת $\frac{1}{2}n(n+1)$ השוואות ו- $\frac{1}{2}n(n+3)$ העתקות.

כל מעבר בלולאת while מבצע העתקה אחת ושתי החלפות, ביחד 7 העתקות. בכל כניסה ללולאה מתבצעת השוואה אחת; אחרי כל מעבר בלולאה מתבצעת עוד השוואה. אם בכניסה ללולאת while האיבר $A[i]$ נמצא כבר במקומו, הלולאה מסתיימת בהשוואה אחת. אחרת, האיבר $A[i]$ ועוד כמה איברים, כמספר המעברים בלולאה, מועברים למקומותיהם; מספר האיברים המועברים למקומותיהם שווה למספר ההשוואות בלולאה. סה"כ מתבצעות בלולאת for n השוואות.

נניח עכשיו שבמערך R נמצאים m אינדקסים i שעבורם $R[i]\neq i$; כלומר, במערך A נמצאים m איברים שאינם במקומותיהם $(0\le m\le n)$. בלולאת for מתבצעות $7m$ העתקות.

האלגוריתם $\text{RANK-SORT1}(A)$ מבצע סה"כ $\frac{1}{2}n(n+3)$ השוואות ו- $(\frac{1}{2}n(n+3)+7m)$ $(0\le m\le n)$ העתקות.

ז'

דוגמה של קלט עבור המקרה הגרוע של האלגוריתם השני הינה $<n,1,2,...,n-1>$; מערך זה משתנה באופן הבא: $<n-1,1,2,...,n-2,n>$, $<n-2,1,2,...,n-3,n-1,n>$, ..., $<2,1,3,...,n>$, $<1,2,...,n>$. כדוגמה של המקרה הגרוע אפשר לקחת את המערך הממוין $<1,2,...,n>$.

ח'

האלגוריתם הראשון מבצע פחות פעולות השוואה; במקרה הגרוע (ובהרבה מקרים אחרים), הוא גם מבצע פחות פעולות העתקה. המצב פחות ברור במקרה הטוב: פחות פעולות השוואה, אבל יותר פעולות העתקה. אפשר לומר שבדרך כלל האלגוריתם הראשון יעיל יותר מבחינת זמן ריצה מאשר האלגוריתם השני.

לעומת זאת, האלגוריתם השני צורך פחות זיכרון; האלגוריתם הראשון זקוק לשלושה מערכים, השני רק לשניים.

ט'

סיבוכיות הזמן של שני האלגוריתמים הינה $\Theta(n^2)$ גם במקרה הטוב וגם במקרה הגרוע. לכן, גם תוחלת זמן הריצה של שני האלגוריתמים $\Theta(n^2)$.