רות הבלין
ת.ז 316243815

ציון

מבחן על-אלגוריתמים

30/40

שאלה 1

אלגוריתם למציאת מסלול בשריג בגודל nxn, מהשכבה השמאלית i=1 לשכבה השמאלית i=n, כאשר המסלול הוא בעל המחיר המזערי:

נבנה מטריצה בגודל nxn (כמו בשריג הנתון, שהעמודות משמאל לימין זה i, והשורות-למעלה למטה - זה j, לשם הנוחות)

עבור כל איבר במטריצה ונחשב מהו הדרך הקצרה ביותר להגיע אליו במטריצה המיוחדת, כאשר נגיע לאיבר מסוים $M(i,j)$ נחשב המחיר להגיע אליו הוא הסכום של האיבר בשריג הנוכחי $c(i,j)$, ועוד אחד האיברים שלפניו $(i-1,j)$ או $(i-1,j+1)$ או $(i-1,j-1)$ המינימלי מביניהם, וזה מה שנמלא ב-$M(i,j)$

כאשר נגיע לשורה ה-$i=n$ ונמצא את כל $j=1..n$, נוכל לדעת היכן המסלול בשריג מסתיים כאשר $M(n,j)$ $1\le j\le n$ הכי קטן.

על מנת להדפיס את המסלול שיש לבדוק כדי להגיע לשם, נחשב שורה אחורה את אותם צעדים שאפשר לבוא ונחפש את האיבר הקטן ביותר, שבדרכו הגענו ורק נשחזר את הצעדים.

לשם השמרה $i=1$ ב-M זה בדיוק $c(1,j)$ $1\le j\le n$ ורק נמלא.

האלגוריתם:

```
min-Route (n)
Array M[1...n , 1...n]
for i=1...n
    for j=1...n
        if i=1   M[i,j] = c(i,j)
        else if j=1       M[i,j] = c(i,j)+min{M[i-1,j], M[i-1,j+1]}
            else if j=n
                   M[i,j] = c(i,j)+min{M[i-1,j], M[i-1,j-1]}
            else
                   M[i,j] = c(i,j)+min{M[i-1,j], M[i-1,j+1], M[i-1,j-1]}
    end for
end for
```

בס"ד

על מנת להדפיס את המסלול נמצא קודם את האיבר המינימלי בשורה $i=n$ ונסמן את האינדקס שבו הוא נמצא כ-k_n, הצעד האחרון בשביל היה למשבצת (n,k_n)
וכך מ $i=n-1$ עד $i=1$ נבדוק עבור כל אחד מ-3 הצעדים המותרים מאיפה הגענו - מאיזה j הגענו, ונסמנו (i, k_i)
כאשר נבדוק ש: $m[i,j] = m[i+1, k_{i+1}] - c(i+1, k_{i+1})$
לאחר שנבדוק את השורה $i=1$, קיבלנו את המסלול ונוכל להדפיסו: $(1, k_1), (2, k_2), \ldots, (n, k_n)$

נכונות האלגוריתם:

בכל שלב כלומר מחשבים את המסלול הקצר ביותר לאיבר (i,j) מסוים, והחישוב מסתמך על מה שחושב ב $i-1$ ובצעדים המותרים בלבד, מלבד בשורה $i=n$ שבה הערך של המסלול הוא המשקל של הנקודה, ולכן הנוסחה בתוך האיטרציה היא מביאה את הפתרון של $OPT(i,j)$
כאשר $OPT(i,j)$ הוא המחיר המינימלי של מסלול מצב שמגיע ל-(i,j)

$$OPT(i,j) = \begin{cases} c(i,j) & i=1 \\ c(i,j) + \min\left\{ \begin{array}{l} OPT(i-1,j), \\ OPT(i-1,j+1), \\ OPT(i-1,j-1) \end{array} \right\} & \text{otherwise} \end{cases}$$

לאחר החישוב, נעבור אחורה מ- $i=n$ ל- $i=1$ באותם צעדים מותרים כמו בחישוב (רק הפוך) ונוציאם את המסלול

לא הוכחה נכונה. צריך להוכיח בצורה יותר מפורטת ופורמלית. למשל באינדוקציה.

זמן ריצה:

באיטרציה רצים על המטריצה m שהוא בגודל $n \times n$ עם ריצה פעולות אלמנטריות ידועות ולכן $\Theta(n^2)$
כדי למצוא את המינימום בשורה ה $i=n$ עוברים על n האיברים ולכן $O(n)$ ואז עוברים מ- $1 \ldots i=n$ ובכל עמודה כלו מבצעים מס' הפעולות אלמנטריות ולכן $\Theta(n)$ ולכן בסה"כ זמן ריצה של האלגוריתם הנ"ל הוא $\Theta(n^2)$ כנדרש.

בס"ד

שאלה 2

אלגוריתם לבניית מגדל יציב בגובה מירבי, שבו קופסא כלפי מעלה קטנה יותר מהקופסאות שמתחתיה באורך וברוחב:

נגדיר את $opt(i)$ להיות הגובה המרבי של מגדל תקין כאשר הקופסא ה-i נמצאת בראש המגדל, מתוך $i=1...n$ תיבות שממוינות לפי אורך התיבה, כך שקופסא $j<i$ אמ"מ $l(j)>l(i)$.

$opt(i)$ הוא מקסימום בין הגובה של התיבה i עצמה לבין הגובה של מגדל תקין ש-i יכול לעמוד מעליהם מצד האורך והרוחב שלו. התיבה i תעמוד לבדה עצמה ו- $opt(i)=h(i)$ רק אם $i=1$ או ש-i היא תיבה שלא יכולה לעמוד מעל תיבות אחרות בגלל גודלה. כמובן שאם התיבה i עומדת מעל מגדל קיים ולא תעמוד לבדה עצמה, נבחר את האפשרות הכי גבוהה של מגדל מבין המגדלים ש-i יכול לעמוד עליהם.

לכן נוכל להגדיר את הנוסחה הבאה:

$$opt(i) = \max\left\{ h(i), \max_{\substack{j<i \\ \& \\ w(j)>w(i)}} \left(opt(j) + h(i)\right) \right\}$$

האלגוריתם יעבוד בצורה איטרטיבית וימלא כל איבר במערך opt, עפ"י הנוסחה הנ"ל, כמו כן ימלא מערך נוסף M שבו בכל איבר $M[i]$ ימלא -1 אם $opt(i)=h(i)$, כלומר המגדל הגבוה ביותר ש-i בראשו הוא רק התיבה ה-i, אחרת $M[i]=j$ כאשר j היא התיבה שמתחת i ומקיימת $\max\limits_{\substack{j<i \\ w(j)>w(i)}} (opt(j))$.

ע"י המערך M נוכל לשחזר את הפתרון ולמצוא אילו תיבות מבין התיבות $1...n$ יש לבנות כדי לקבל מהם גובה מקסימלי, נמצא את האיבר המקסימלי במערך opt, נניח שזה האיבר ה-i ואז נגיד ש-i בראש המגדל ואז נמצא את $M[i]=j$ והוא יהיה מתחתיו ואז נלך ל- $M[j]$ וכך עד שנקבל -1 ונדע שאיבר זה הוא הבסיס של המגדל.

המשך

האלגוריתם -

- נמיין את n התיבות כאשר $1 \le i < j \le n$ ומתקיים $l(i) > l(j)$
- $opt(1) = h(1)$
- for $i = 2 \ldots n$ נמלא את $opt(i)$ עפ"י הנוסחה,
$$opt(i) = \max\{h(i), \max_{\substack{j<i \\ \& \\ w(j)>w(i)}} (opt(j) + h(i))\}$$

ונשמור - $m[i] = j$ (ה-j שנמצא ש $opt(j)$ הוא ה-max)

- נחפש את $\max_{1 \le j \le n}(opt(j))$, ונקרא $k \leftarrow j$

- נדפיס שהתיבה k במגדל do →
  while → $m[k] \neq -1$
  $k \leftarrow m[k]$

(הדפסת התיבות שנמצאות במגדל היא מלמעלה ללמטה, התיבה הכי למעלה תודפס קודם, והתיבה שהיא הבסיס תודפס אחרונה, אפשר בקלות להפוך אם רוצים...)

נכונות האלגוריתם - הסברתי את נוסחת הנסיגה בהסבר לפני האלגוריתם.

זמן ריצה - מיון התיבות $\Theta(n\log(n))$
מילוי המערך - opt הוא $\Theta(n^2)$ כי עבור כל איבר i בודקים מה המקסימום של $opt(j)$ עבור כל $j<i$. חיפוש המקסימום במערך opt הוא $\Theta(n)$
הדפסת מספרי התיבות ע"י המערך m היא כמובן גם $\Theta(n)$
ולכן סה"כ זמן ריצה של האלגוריתם $\Theta(n^2)$ כנדרש.

בס"ד

8/20

שאלה 4

א) האלגוריתם שמוצג בשאלה מחשב את המסלול הקל המשקל המינימלי של כל הקשתות שעוברות בו, מקודקוד r לקודקודים האחרים בגרף, האלגוריתם ממלא את המערך A, באיבר ה-A[i] לאחר סיום האלגוריתם 'יופיע' המשקל של המסלול המינימלי מהקודקוד r לקודקוד i, אם אין מסלול מ-r לקודקוד V_i אז $A[i]=\infty$.

<u>הוכחה באינדוקציה:</u>

נוכיח על מספר האיטרציות של הלולאה החיצונית.

<u>טענה</u> - בסיום כל איטרציה i, כל איבר ב-A יכיל את המשקל של המסלול המינימלי מ-r ל-v מבין המסלולים שנסרקו עד

האלגוריתם לא סורק מסלולים אלא עובר על קשתות ולכן הטענה לא מוגדרת היטב.

כה ב-i האיטרציות, במידה ולא נסרק מסלול כזה אז $A[v]=\infty$.

<u>בסיס האינדוקציה</u> - לפני האיטרציה הראשונה של הלולאה החיצונית המערך A מלא בכל $A[v]=\infty$ חוץ מ-v $A[v=r]=0$ ואכן המשקל של המסלול הקל ביותר מ-r לעצמו הוא כמובן 0, כל שאר הצמתים טרם נסרק מסלול מ-r אליהם ולכן משקלם ∞.

<u>צעד האינדוקציה</u> - נניח נכונות עבור האיטרציות $n\geq i$ ונניח עבור האיטרציה $n=i$. בכל איטרציה נסרקים כל צלעות הגרף בסדר לקסיקוגרפי, המסלול הארוך ביותר שאפשרי שיהיה ושהוא ~~גם~~ גם המסלול הקל ביותר בין n הקודקודים בגרף יהיה לכל היותר עם n-1 צלעות. לכן למקרה הגרוע (בגלל הסריקה הלקסיקוגרפית) ויתכן ובכל איטרציה i של הלולאה החיצונית יסרק המסלול בעל המשקל המינימלי ביותר מ-r לצומת v שהוא במרחק i צלעות בדיוק בה מצב ויעודכן ב A[v] להיות המינימלי. בכל איטרציה אחרת באיטרציה i יסרקו מסלולים מינימלים מ-r ל-v שמרחקיהם בצלעות קטן מ-i. לכן חייב להיות שבסוף של איטרציה ה-i כאשר $n=i$ כל המסלולים שהם במרחק i צלעות לכל היותר לפי סדר סריקה לקסיקוגרפי נסרקו כבר וכבר נמצאו המסלול המצאו' מבניהם. לכן בסריקה ה $n=i$ כבר לא ישתנה כלום במערך A במהלך האיטרציה, ולכן האלגוריתם יסיים וקיבלנו את המשקלים המינימליים של מסלולים מ-r לשאר הקודקודים.

לא הוכחה נכונה (בעיה בטענת האינדוקציה). -10

בס"ד

ד) לפי ההוכחה בסעיף א' המספר המרבי של האיטרציות
הדרושה הינה על גרפים בעלי n קודקודים היא n איטרציות

לא ברור למה זה נובע מההוכחה של סעיף א'. אני לא השתכנעתי -2

ולכן $B(n)=n$, סדרת גרפים שמקיימת בדיוק n איטרציות
כשנפעיל את האלגוריתם הנ"ל היא כאשר הקודקודים ממוספרים
מ-1 עד n- כאשר הקודקוד $r=n$, ובכל קודקוד i
יש ממנו צלע מכוונת לכוון הקודקוד ה $i-1$, כאשר לקודקוד
$i=1$ אין צלע שיוצאת ממנו אליו רק שנכנסת. דוגמה הבין:
(כאשר לא משנה מה המשקל של הצלעות מלבד שיהיו אי שליליים)

$$1 \leftarrow 2 \leftarrow 3 \leftarrow \cdots \leftarrow n-1 \leftarrow n \ \ r$$

ה) סעיף
סדרת גרפים שונה, שבה הקודקודים ממוספרים
מ-1 עד n והקודקוד $r=1$ הצלעות מכוונות
מכל קודקוד i, לקודקוד הבא אחריו לפי הסדר הלקסיקוגרפי -
$i+1$, כך ; (המשקלים לא משנים, רק שיהיו אי שליליים)

$$r \ \ 1 \rightarrow 2 \rightarrow 3 \rightarrow \cdots \rightarrow n-1 \rightarrow n$$

ואז למעשה כבר באיטרציה הראשונה הקודקוד 2
יתעדכן להיות במערך, כערך של הקשת $(1,2)$ פלוס $A[1]$
שהוא 0, ואז יחושב $A[3]$ וכו עפ"י הסדר הלקסיקוגרפי,
ובסוף האיטרציה ה $i=1$ כל המערך מלא במשקל של
המסלול מ-r לכל הקודקודים בגרף.
באיטרציה השנייה יעבור שוב על כל הגרף, אך שום
דבר לא ישתנה ולכן האלגוריתם יסתיים.
כך קיבלנו אותו מס' צלעות בגרף אך 2 איטרציות
בלבד במקום n איטרציות.