

שאלה 1 (40 נק')

א. (25 נק') יהי $G(V, E)$, גרף קשיר , $w: E \rightarrow \mathbb{R}_+$ משקל חיובי.
יהי $\mathcal{T}$ מרחב כל העצים הפורשים של G ויהי $T \in \mathcal{T}$ עץ פורש.
יהי $T' \in \mathcal{T}$ עץ פורש מינימלי (MST).

נסמן $w^*(T) = \max_{e \in E_T} w(e)$ (המשקל המקסימלי של צלע ב T) וכן

$T'' = \operatorname{argmin}_{T \in \mathcal{T}} w^*(T)$ (כאשר T'' העץ שבו לצלע המקסימלית יש משקל מינימלי – עץ צוואר בקבוק)

הוכיחו שהעץ הפורש המינימלי T' הוא גם עץ צוואר בקבוק T'' .

הנחיה לפתרון:

הניחו ש MST איננו עץ צוואר בקבוק.
הראו כיצד למצוא צלע בעץ צוואר בקבוק שיכולה להחליף צלע של MST .

הוכחה

נניח בשלילה ש T' איננו עץ צוואר בקבוק.

יהיו $e^* = \operatorname{argmax}_{e \in T'} w(e)$ ו $e^{**} = \operatorname{argmax}_{e \in T''} w(e)$ הצלעות בעלות המשקל המקסימלי ב T' ו T'' בהתאמה.

מתקיים $w(e^{**}) < w(e^*)$ כי אחרת T' היה עץ צוואר בקבוק בניגוד להנחה.

ולכן מתקיים $\forall e \in T'', w(e) \leq w(e^{**}) < w(e^*)$

מחיקת $e^*(u, v)$ מותירה שני תתי עצים $T'(V_u, E_u)$ ו $T'(V_v, E_v)$, $V = V_u \cup V_v$.

נתבונן בחתך $[V_u, V_v]$ ובפרט בצלע $e^* \in [V_u, V_v]$.

מאחר ו T'' קשיר, קיימת $e^{***} \in T''$, $e^{***} \in [V_u, V_v]$.

$e^{***} \notin T'$ כי אחרת ב T' היה מעגל דרך e^* ו e^{***} .

מאחר ו $w(e^{***}) < w(e^*)$, הוספת e^{***} ל T' ומחיקת e^* תיצור עץ פורש שמשקלו קטן יותר, סתירה לכך ש T' הוא MST . ■

ב. (15 נק') בהנחה שמשקלי הצלעות כולם שונים זה מזה. הוכיחו ש MST הוא יחיד.

הנחיה לפתרון: הניחו שקיימים שני MST והתבוננו בצלע המינימלית בקבוצת הצלעות הזרות שלהם, כלומר הצלעות השייכות לעץ אחד אבל לא לשני ולהיפך.

הוכחה

נניח בשלילה שקיימים $T_1(E_1, V)$ ו $T_2(E_2, V)$ שני MST שונים זה מזה.

נתבונן ב $E_3 = (E_1 \cup E_2) \setminus (E_1 \cap E_2)$ הצלעות הזרות שלהם. תהי $e^* = \operatorname{argmin}_{e \in E_3} w(e)$

הצלע המינימלית, ונניח בלי הגבלת הכלליות ש $e^* \in E_1, e^* \notin E_2$.

הוספת e^* ל T_2 יוצרת מעגל C . במעגל C קיימת צלע $e^{**} \in E_2 \cap E_3$ כלומר צלע שאיננה ב E_1 כי אחרת היה ב T_1 מעגל מלכתחילה, דבר בלתי אפשרי.

מאחר ומשקל כל הצלעות שונה, מאופן בו נבחר e^* מתקיים $w(e^*) < w(e^{**})$.

ולכן $T_2 \cup e^* \setminus e^{**}$ הוא עץ פורש שמשקלו קטן ממשקל T_2 בסתירה להיותו MST. ■

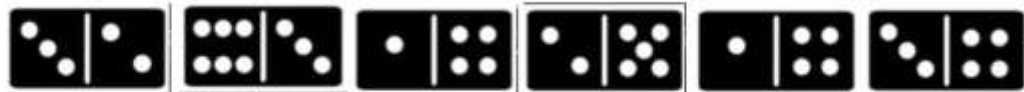
שאלה 2 (40 נק')

נתונים n אבני דומינו מסודרים בשורה שערכיהן $(X_i, Y_i), 1 \leq i \leq n$.

את כל אחת מהאבנים $1 \leq i \leq n$ ניתן לסובב. ונסמן את ערכי האבן (L_i, R_i) על פי הסיבוב: במצב אחד $L_i = X_i$ ו $R_i = Y_i$ ובמצב השני $L_i = Y_i$ ו $R_i = X_i$.

נדרש למצוא סיבוב של האבנים שעבורו הסכום $C = \sum_{i=1}^{n-1} R_i \times L_{i+1}$ מקסימלי. בדוגמא שלהלן $n = 6$ ומוצגות שתי אפשרויות שונות לסיבוב האבנים.

$$\sum_{i=1}^5 R_i \times L_{i+1} = 2 \times 6 + 3 \times 1 + 4 \times 2 + 5 \times 1 + 4 \times 3 = 41$$



$$\sum_{i=1}^5 R_i \times L_{i+1} = 3 \times 3 + 6 \times 4 + 1 \times 2 + 5 \times 4 + 1 \times 4 = 59$$



א. (5 נק') מהו גודל מרחב הסידורים האפשריים? הסבירו במדויק את תשובתכם.
מאחר וישנן n קוביות וכל אחת יכולה להיות בשני מצבים אזי מרחב הפתרונות הוא בגודל 2^n .

הציגו אלגוריתם יעיל ככל האפשר שמחשב את C המקסימלי בסיבוכיות זמן חסום ע"י פולינום ע"פ הסעיפים שלהלן. הנחיה לפתרון:

סמנו ב C_i^Y את הערך המקסימלי האפשרי לסיבוב האבנים 1 עד i כאשר $R_i = Y_i$ וב C_i^X את הערך המקסימלי האפשרי לסיבוב האבנים 1 עד i כאשר $R_i = X_i$.

הניחו שערכי קוביות הדומינו (X_i, Y_i) , $1 \leq i \leq n$ נתונים בשני מערכים $X[n]$ ו $Y[n]$.

ב. (15 נק') הסבירו בפירוט ובמדויק את האלגוריתם שבחרתם וכיצד מחושב

$$C = \sum_{i=1}^{n-1} R_i \times L_{i+1}$$

ניתן לפתור בעזרת אלגוריתם תכנות דינאמי. בסימון הנ"ל מתקיים

$$C_n^Y = \max\{C_{n-1}^Y + Y_{n-1} \times X_n, C_{n-1}^X + X_{n-1} \times X_n\}$$

$$C_n^X = \max\{C_{n-1}^Y + Y_{n-1} \times Y_n, C_{n-1}^X + X_{n-1} \times Y_n\}$$

ג. (10 נק') הציגו pseudo code למימוש האלגוריתם.

אפשר לפתור באופן רקורסיבי מהסוף להתחלה בעזרת הקוד שלהלן

$$C = \text{MAX_CHAIN}(n, X, Y; CY, CX)$$

$$\text{MAX_CHAIN}(n, X, Y; CY, CX)$$

$$\text{MAX_CHAIN}(n - 1, X, Y; CY_old, CX_old);$$

$$CY = \max\{CY_old + Y[n - 1] \times X[n], CX_old + X[n - 1] \times X[n]\};$$

$$CX = \max\{CY_old + Y[n - 1] \times Y[n], CX_old + X[n - 1] \times Y[n]\};$$

$$\text{return max}\{CY, CX\}$$

ד. (10 נק') נתחו במדויק את סיבוכיות זמן הריצה של האלגוריתם.

הפונקציה הרקורסיבית נקראת סה"כ n פעמים ובכל קריאה מבצעת מספר קבוע של פעולות סקלריות ולכן $T(n) = \Theta(n)$.

שאלה 3 (40 נק')

נתונות n נקודות במישור (p_i, q_i) , $1 \leq i \leq n$. שתי נקודות (p_i, q_i) ו (p_j, q_j) , $i \neq j$, הן זהות אם ורק אם $p_i = p_j$ וגם $q_i = q_j$, אחרת הן שונות. מניחים שכל n הנקודות שונות.

הסדר בין הנקודות מוגדר כדלהלן: $(p_j, q_j) > (p_i, q_i)$ אם $p_j > p_i$ או $p_j = p_i$ וגם $q_j > q_i$.

שני מהנדסים בחברת "הנקודה העיקרית" נדרשו לארגן את הנתונים בעץ חיפוש בינארי (BST).

מהנדס א' החליט להשתמש ב BST דו-מימדי: BST ראשי שהצמתים שלו מאכסנים את ערכי p . כל צומת ב BST הראשי מצביע ל שורש של BST משני שבצמתים שלו מאוכסנים הערכים השונים של q עבור p זהה.

ענו על השאלות שלפניכם בצורה מדויקת ונמקו היטב כל תשובה.

א. (5 נק') מהי סיבוכיות גודל הזכרון הנדרש לאכסון הנתונים (סיבוכיות המקום)?

כל נקודה תופסת שני צמתים, האחד בעץ הראשי והשני בעץ המשני. לכן סיבוכיות גודל הזכרון $\Theta(n)$

ב. (5 נק') מהי סיבוכיות זמן החיפוש לנקודה במקרה הגרוע ביותר, ומתי הוא קורה?

במקרה הגרוע ביותר מסלול החיפוש הוא רשימה ליניארית של $n + 1$ צמתים. עץ כזה נוצר כאשר בנקודות המוכנסות לעץ הדו-מימדי כל ערכי q שונים זה מזה או כאשר כל ערכי p שווים אבל כל ערכי q שונים, וסדר ההכנסה של הנקודות מהקטן לגדול או להיפך.

מהנדס ב' אימץ את רעיון העץ הדו-מימדי של מהנדס א' אבל החליט להשתמש בעצי AVL.

ג. (5 נק') מהי סיבוכיות גודל הזכרון הנדרש לאכסון הנתונים?

כמו א'.

ד. (5 נק') מהי סיבוכיות זמן החיפוש לנקודה במקרה הגרוע ביותר, ומתי היא קורית?

מאחר ועומק עץ AVL $\Theta(\log_2 n)$ והדבר נכון הן לעץ הראשי והן לעצים המשניים אזי אורך מסלול החיפוש במקרה הגרוע ביותר $\Theta(\log_2 n)$. הדבר קורה תמיד, ללא תלות בסדר הכנסת הנקודות.

ה. (15 נק') נדרש להדפיס את ערכי הנקודות (p_i, q_i) , $1 \leq i \leq n$ המאוכסנות בעץ הדו-מימדי בסדר ממזין (אמור לעבוד ללא הבדל במימוש BST או AVL).

הציגו pseudo code מתאים.

הנחיה לפתרון אפשרי: שימוש ברקורסיה למעבר על צמתי העצים.

העזרו בסימונים הבאים:

x – מצביע לצומת בעץ הראשי, y – מצביע לצומת בעץ המשני

$x.y$ – מצביע מצומת בעץ הראשי לשורש בעץ המשני.

$x.left, x.right, y.left, y.right$ – מצביעים לבן ימני ושמאלי בעצים המתאימים

$x.p, y.q$ – ערכי המפתחות (קואורדינטות) המתאימים לנקודה (p, q)

משתמשים בשתי סברוטינות IN ORDER למעבר על צמתי העץ הראשי והעץ המשני. כל צומת

בעץ הראשי קוראת לסברוטינות IN ORDER בעץ המשני המתאים לה, ומעבירה את המצביע x

שבו נמצא ערך p המשותף.

```
INORDER-XTREE( $x$ )  
  if  $x \neq NIL$   
    INORDER-XTREE( $x.left$ )  
    INORDER-YTREE( $x, x.y$ )  
    INORDER-XTREE( $x.right$ )
```

```
INORDER-YTREE( $x, y$ )  
  if  $y \neq NIL$   
    INORDER-YTREE( $x, y.left$ )  
    print( $x.p, y.q$ )  
    INORDER-YTREE( $x, y.right$ )
```

ו. (5 נק') מהי סיבוביות הזמן של קוד ההדפסה שמימשתם? רישמו הסבר מפורט.

סיבוכיות זמן הריצה ליניארית. לכל צומת בעץ הראשי והעצים המשניים מגיעים פעמיים בדיוק.
פעם בכניסה לרקורסיה המתאימה ופעם ביציאה ממנה.