

סוג הבחינה: מבחן מתכונת
מבחן: מתכונת 8
מספר השאלון: 899271
קובץ: תשובות סופיות

מדעי המחשב

מבני נתונים ומסלול - 4-5 יחידות לימוד

הוראות

- א. משך הבחינה: שלוש שעות.
 - ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים.
פרק ראשון - 2×25 - 50 נקודות
פרק שני - 2×25 - 50 נקודות
סך הכול - 100 נקודות
 - ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשבון שיש בו אפשרות תכנות.
 - ד. הוראות מיוחדות:
 - (1) רשמו על הכריכה החיצונית של המחברת את שם המסלול שלמדתם.
 - המסלול הוא אחד משלושת המסלולים האלה: אלגוריתמים, מודלים חישוביים, תכנות מונחה עצמים.
 - (2) את כל התוכניות שיש לכתוב בשפת מחשב בפרקים הראשון והשני כתבו בשפה אחת בלבד - Java או C#.
 - הערה: לא יורדו נקודות אם תכתבו בתוכניות אות גדולה במקום אות קטנה או להפך.
 - יש לכתוב במחברת הבחינה בלבד. יש לרשום "טיוטה" בראש כל עמוד המשמש טיוטה.
 - כתיבת טיוטה בדפים שאינם במחברת הבחינה עלולה לגרום לפסילת הבחינה.
 - השאלות בשאלון זה מנוסחות בלשון רבים, אף על פי כן על כל תלמידה וכל תלמיד להשיב עליהן באופן אישי.
- בהצלחה!**
- השאלות
- בשאלון זה שני פרקים. יש לענות על שאלות משני הפרקים, לפי ההוראות בכל פרק.
- הערה: בכל שאלה שנדרשת בה קליטה, אין צורך לבדוק את תקינות הקלט.
- לפותרים בשפת Java: בכל שאלה שנדרשת בה קליטה, הניחו שבתוכנית כתובה ההוראה:
- Scanner input = new Scanner (System.in);
- שימו לב: בכל שאלה שנדרש בה מימוש אפשר להשתמש בפעולות של המחלקות: תור, מחסנית, עץ בינרי וחוליה, בלי לממש אותן. אם משתמשים בפעולות נוספות, יש לממש אותן.

תשובות סופיות

פרק ראשון

שאלה 1

א(1). $\text{ppp}(q, 4)$ מחזירה false ($c = 3$): האיברים 4, 7, 9 גדולים מכל האיברים שלפניהם; התור חוזר למצבו המקורי.

א(2). מחזירה true אם בתור יש לפחות k איברים שכל אחד מהם גדול מכל האיברים שלפניו בתור ("שיאים"; האיבר שבראש התור תמיד נספר), אחרת false; התור נשאר ללא שינוי.

א(3). $O(n)$ - סיבוב אחד של התור: n סיבובי לולאה, $O(1)$ בכל סיבוב.

ב(1). $\text{find}(q, 1)$ מחזירה 3 ($k = 1, 2, 3 - \text{true}; k = 4 - \text{false}$) מוחזר $3 = 1 - 4$.

ב(2). $\text{find}(q, 1)$ מחזירה את מספר האיברים בתור שגדולים מכל האיברים שלפניהם (מספר ה"שיאים"); התור לא משתנה.

ב(3). $O(n^2)$ במקרה הגרוע: עד $n + 1$ קריאות רקורסיביות, בכל אחת ppp ב- $O(n)$.

שאלה 2

א. pv, pj על החוליה הראשונה; כל עוד $\text{count} < \text{steps}$ וגם $\text{count} \neq \text{pv} \neq \text{null}$: $\text{sum} += \text{pv}.\text{getValue}()$, $\text{count}++$, ואז מקדמים את pv ב- $\text{pj}.$ במקביל $\text{jump} = \text{pj}.\text{getValue}()$ פעמים (עוצרים אם pv נהיה null).

ב. pv, pj עוברים במקביל על כל המיקומים; לכל מיקום: אם $\text{collect}(\text{pv}, \text{pj}, \text{steps}) == \text{target}$ מחזירים true; בסוף false.

שאלה 3

עלה $\leftarrow \text{true}$. בצומת אחר: אם ערכו שונה מ- false $\leftarrow \text{sumTree}(\text{left}) + \text{sumTree}(\text{right})$; אחרת בודקים ברקורסיה כל בן קיים. בדוגמה הראשונה מתקבל true, ובשנייה false (בצומת 5: $6 = 4 + 2$).

פרק שני (50 נקודות) - אלגוריתמים

שאלה 4

א. BFS מ- v_s שבו לכל קודקוד שמתגלה שומרים את ה"הורה" $\pi[v]$ (הקודקוד שממנו התגלה). אם v_t לא התגלה - מחזירים רשימה ריקה; אחרת הולכים מ- v_t אחורה לפי π עד v_s ומחזירים את המסלול בסדר הפוך. זמן ריצה $O(|V| + |E|)$. בדוגמה: 2, 4, 9.

ב. עוברים על הקודקודים לפי הסדר; מכל קודקוד שעוד לא סומן מריצים BFS שאוסף את הרכיב ומסכם את דרגות הקודקודים שלו. מספר הקשתות ברכיב = סכום הדרגות / 2. שומרים את הרכיב עם המספר הגדול ביותר. זמן ריצה $O(|V| + |E|)$. בדוגמה: {2, 4, 6, 9, 10} עם 6 קשתות.

שאלה 5

א. בונים את הגרף ההפוך (כל קשת $u \rightarrow v$ הופכת ל- $v \rightarrow u$) ב- $O(|V| + |E|)$, ומריצים BFS (או DFS) מ- v_j . כל קודקוד שמתגלה (חוץ מ- v_j) מוחזר. זמן ריצה $O(|V| + |E|)$. בדוגמה מתגלים 2, 3, 8 ואחר כך 0.

- ב. מיון טופולוגי, ואז מעבר על הקודקודים לפי הסדר: $L[v] =$ האורך המקסימלי של שרשרת שמסתיימת ב- v $L[v] = \max(L[u] + 1)$ על כל קשת $u \rightarrow v$, עם שמירת הקודם $P[v]$. מחזירים את השרשרת מהקודקוד שבו L מקסימלי אחורה לפי P . זמן ריצה $O(|V| + |E|)$. בדוגמה: 0, 2, 5, 1, 7, 4.

שאלה 6

- א. למשל: קודקודים 1-6, קשתות 1-2, 2-3, 3-4, 4-5, 5-6. אין קשת בין 3 ל-4, והוספת הקשת 3-4 נותנת עץ.
 ב(1). כן. מספר הקשתות הוא $n-2$.
 ב(2). כן. יש בדיוק 2 רכיבי קשירות.
 ג(1). G הוא גרף-עץ-שבור $\Leftrightarrow$ יש בו בדיוק $n-2$ קשתות ובדיוק 2 רכיבי קשירות. האלגוריתם: סופרים קשתות (עוצרים אם יש יותר מ- $n-2$); אם $n-2 \neq$ שקר; אחרת סופרים רכיבי קשירות בעזרת BFS מכל קודקוד שעוד לא התגלה: אם התקבלו בדיוק 2 - אמת, אחרת שקר.
 ג(2). $O(n)$ (ובאופן כללי $O(V+E)$, אבל בזכות העצירה המוקדמת $n-2 \leq E$).

פרק שני (50 נקודות) - מודלים חישוביים

שאלה 7

- א(1). L_1 - לא רגולרית; L_2 - רגולרית (3 מצבים); L_3 - רגולרית.
 א(2). לא. $L_1 \cap \bar{L}_3$ סופית, ולכן אילו $L_1 \cap L_3$ הייתה רגולרית, $L_1 = (L_1 \cap L_3) \cup (L_1 \cap \bar{L}_3)$ הייתה רגולרית - סתירה.
 א(3). כן. $\bar{L}_3 =$ כל המילים באורך קטן מ-12, ולכן $L_1 \cap \bar{L}_3$ סופית - וכל שפה סופית רגולרית.
 א(4). לא. $L_1 \subseteq L_2$, ולכן $L_1 = L_2 \cap (L_1 \cap \bar{L}_1)$. (משלים של $(L_2 \cap \bar{L}_1)$). אילו $L_2 \cap \bar{L}_1$ הייתה רגולרית, L_1 הייתה רגולרית (משלים וחיתוך) - סתירה.
 ב(1). $ba\#b$ - שייכת; $ab\#aab$ - שייכת; $ab\#ab$ - לא שייכת; $a\#a$ - לא שייכת; $aa\#b\#a$ - לא שייכת.
 ב(2). $p_0 \rightarrow p_2: b$; $p_1 \rightarrow p_1: a$; $p_1 \rightarrow p_2: b$; $p_2 \rightarrow p_1: a$; $p_3 \rightarrow p_4: a, b$; $p_2 \rightarrow p_5: \#$; $p_6 \rightarrow p_7: a, b$; $p_7 \rightarrow p_8: a, b$; $p_8 \rightarrow p_6: a, b$. מצבים מקבלים: p_4, p_8 .

שאלה 8

- א(1). $L_4 \cap L_5 = \{xy\}$
 א(2). $L_6 \cdot L_6 = \{yy, yyx, yxy, yxyx\}$
 א(3). $L_1 \cdot L_2 = \Sigma^* = L_1$
 א(4). $L_3^R = \emptyset$
 א(5). $L_5 \cdot L_6 = \{y, yx, xy, xyx, xyy, xyyx\}$
 ב. דוחפים סימן לכל a ראשון ולכל b ; על כל c שולפים סימן של b , ואחר כך על כל a שבסוף שולפים סימן של a . הסימנים התחתונים של כל בלוק מסומנים (A - ה- a הראשון; Z/Y - ה- b הראשון כש- $n \geq 0$; $n = 0$) כדי לדעת בלי מעברי ε מתי בלוק נגמר. מצב מקבל יחיד: q_5 .

שאלה 9

- א(1). 7 שלבים (0-6), מסתיים ב- q_6 ; המילה מתקבלת והסרט בסוף: $\vdash b b \Delta \dots$
 א(2). 9 שלבים (0-8), מסתיים ב- q_6 ; המילה מתקבלת והסרט בסוף: $\vdash a b a \Delta \dots$

א(3). $\vdash \Delta b a b a \dots$ (על הסרט: babab)

ב(1). 8 מילים - בדיוק המילים שהאות האחרונה בהן שווה לאות הראשונה ($8 = 2 \cdot 2 \cdot 2$).

ב(2). המכונה מחליפה את האות האחרונה במילה בעותק של האות הראשונה, וחוזרת לתחילת הסרט. היא מקבלת כל מילה לא ריקה ודוחה את המילה הריקה.

תכנות מונחה עצמים

שאלה 10

א(1). Cat, Dog ו-Rabbit יורשות מ-Pet (חץ ירושה אל Pet); Pet מכילה עצם מטיפוס Stay (הכלה, המעוין בצד של Pet).

א(2). Stay: owner, nights, meals; Pet: chip, stay; Cat extends Pet: indoor; Dog extends Pet: weight; Rabbit extends Pet: age.

ב(1). `this.stay = new Stay(owner, nights, meals); this.chip = chip`

ב(2). Java: `super(owner, nights, meals, chip); this.weight = weight`; C#: `base(owner, nights, meals, chip) { this.weight = weight; }`

ג(1). C#: `public virtual double`; Java: `return 90 * this.stay.getNights() + 12 * this.stay.getMeals` (Cost עם GetNights/GetMeals)

ג(2). Cat - אין צורך. Dog: אם `weight > 20` מחזירה `getNights().getStay() + 15 * super.cost()`, אחרת `base.Cost()` ו-C#: `override; super.cost();` Rabbit: `return super.cost() * 0.75`

שאלה 11

א. Box(): `size=1`; Box(int size): `made++`, `size=size`; GiftBox(String color): `super(color.length())`; color=color; GiftBox(int size, String color): `super()`; `this.size=size`; `color=color`; `toString(): super.toString() + ", color=" + color`

ב(1). 1 - לא ייתכן; 2 - ייתכן; 3 - לא ייתכן; 4 - ייתכן; 5 - לא ייתכן; 6 - ייתכן.

ב(2). כן. ההמרה $k3(L)$ של עצם M מצליחה רק אם M יורשת מ-L, ולכן רק תרשים 2 נשאר אפשרי (ב-4 וב-6 תהיה שגיאת ריצה).

ג. protected boolean broken; protected static double rate; public Device(); public Device(String code, int power); public String info (ב-C#: `public virtual string Info()`); public boolean isOld(int year) (ב-C#: `public bool IsOld(int year)`)

שאלה 12

א. Robot Alfa / Robot Beta / Cleaner / Robot Gama / Cleaner 60 / Robot Delta / Cleaner r1: עצמים: Robot(Alfa,100), r2 Cleaner(Beta,100,0), c1 Cleaner(Gama,60,0), c2 Cleaner(Delta,100,0); made = 4

ב. 1 תקין - 2 | 4 Alfa(100) תקין - 3 | Beta worked, energy = 90 לא תקין (הידור: אין work(int,int)
ב-Robot) | 4 תקין - 5 | 6 rooms = 30 / energy = 6 | 0 rooms: 0 | 6 Beta(100) לא
תקין (הידור: Robot ל-Cleaner בלי המרה) | 7 לא תקין (שגיאת ריצה - המרה לא חוקית) | 8 לא תקין
(הידור: אין פעולה בונה עם 3 פרמטרים) | 9 תקין - 10 | 2 rooms = 90 / energy = 2 Beta worked, energy = 90 -
Delta worked, energy = 80

כל הבגרויות במדעי המחשב עם פתרונות מלאים: www.guyai.co.il/bagrut/computer-science