

סוג הבחינה: בגרות
מועד הבחינה: קיץ תשפ"ד, 2024
מספר השאלון: 899271
קובץ: תשובות סופיות

מדעי המחשב

מבני נתונים ומסלול - 4-5 יחידות לימוד

הוראות

- א. **משך הבחינה:** שלוש שעות וחצי.
- ב. **מבנה השאלון ומפתח ההערכה:** בשאלון זה שני פרקים.
יש לבחור בארבע שאלות סך הכול. לכל שאלה - 25 נקודות; סך הכול - 100 נקודות.
שימו לב: אם תבחרו לענות על שאלות מן הפרק השני, בחרו בשאלות מתוך מסלול אחד בלבד.
ג. **חומר עזר מותר בשימוש:** כל חומר עזר, חוץ ממחשבון שיש בו אפשרות תכנות.
- ד. **הוראות מיוחדות:**
 - (1) אם תבחרו לענות על שאלות רק מן הפרק הראשון **רשמו על הכריכה החיצונית** של המחברת "**ללא מסלול**", אחרת רשמו 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

- יש לענות על ארבע שאלות סך הכל מן הפרקים הראשון והשני (לכל שאלה - 25 נקודות).
- 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. מעבירים את כל איברי התור לתור עזר תוך ספירת המקום; שומרים את האיבר במקום $m-1$ ואת האיבר במקום m , וכשמגיעים למקום $m+1$ בודקים `before + x == cur` (רק אם $m > 1$). בסוף מחזירים את האיברים לתור. אם $m = 1$ או m הוא המקום האחרון - `false`.
- ב. סופרים את גודל התור (ומחזירים את האיברים לתור), ואז בודקים `isMagic(q, m)` לכל $m = n, 2n, 3n$, $m = n$, $m = 2n$, $m = 3n$... גודל התור. אם אחד מהם אינו איבר קסם - `false`, אחרת `true`.

שאלה 2

- א. מעבירים את q לתור עזר. לפני החולה הראשון שרמת הדחיפות שלו **נמוכה ממש** מזו של p מכניסים את p . אם אין חולה כזה, p נכנס בסוף. אחר כך מחזירים את כל האיברים מתור העזר ל- q .
- ב. מוציאים את החולה עם מספר הזהות `id` מהתור (שאר החולים נשארים באותו סדר), מעדכנים את רמת הדחיפות שלו: `setPriority(pri)`, ומכניסים אותו מחדש עם `priorityInsert` מסעיף א.

שאלה 3

- א(1). הפעולה מחזירה את השרשרת $3 \rightarrow 5 \rightarrow 3 \rightarrow 9 \rightarrow 4 \rightarrow \text{null}$ (שתי החוליות עם הערך 1 הוסרו).
- א(2). הפעולה מוחקת מהשרשרת `lst` את כל החוליות שהערך שלהן שווה ל- x (סדר שאר החוליות נשמר) ומחזירה הפניה לראש השרשרת החדשה (`null` אם לא נשארו חוליות).
- א(3). $O(n)$, כאשר n הוא מספר החוליות בשרשרת: קריאה רקורסיבית אחת לכל חוליה (ועוד אחת ל-`null`), ובכל קריאה עבודה קבועה.
- ב(1). בסיום הפעולה: `lst = 1 \rightarrow 3 \rightarrow 5 \rightarrow 9 \rightarrow 4 \rightarrow \text{null}`.
- ב(2). הפעולה מוחקת מהשרשרת את הכפילויות: מכל ערך נשארת רק ההופעה הראשונה שלו (סדר ההופעות הראשונות נשמר).
- ב(3). $O(n^2)$, כאשר n הוא מספר החוליות: לכל חוליה מופעלת `what` על ההמשך שלה, שאורכו עד $n-1, n-2, \dots, 0$. הסכום הוא $n(n-1)/2$.

שאלה 4

- א. חיפוש סדרתי במערך `arr` בתאים 0 עד `amount-1`: אם `arr[i] == n` מחזירים `true`. בסוף הלולאה מחזירים `false`.
- ב. עוברים על הקווים של התחנה הראשונה `arr[0]`. לכל קו בודקים בעזרת `isStopping` אם הוא עוצר בכל שאר התחנות. אם כן - מוסיפים אותו כחוליה חדשה לראש השרשרת. בסוף מחזירים את השרשרת.

שאלה 5

- א. עוברים פעם אחת על השרשרת וסופרים מקומות. שומרים את המקום של `posNum` ואת המקום של `posOpp`. אם `num` לא נמצא - מחזירים -1, אחרת מחזירים $|\text{posNum} - \text{posOpp}| + 1$.

- ב. $\max = -1$. לכל ערך v בשרשרת מחשבים $\text{width}(\text{lst}, v)$ ומעדכנים את $\max$ אם התוצאה גדולה יותר.
מחזירים את $\max$ (נשאר -1 אם אין אף תת-סדרה נגדית).

פרק שני – אלגוריתמים

שאלה 6

- א. סריקת BFS (או DFS) מ- v_j : כל קודקוד שמתגלה בסריקה (חוץ מ- v_j עצמו) מוחזר. זמן ריצה $O(|V| + |E|)$. בדוגמה מ-3 מתגלים 0, 7, 9, 6.
ב. עוברים על הקודקודים $v_0 \dots v_{n-1}$; מכל קודקוד שעוד לא סומן מריצים BFS שאוסף את רכיב הקשירות שלו; שומרים את הרכיב בעל מספר הקודקודים הקטן ביותר. זמן ריצה $O(|V| + |E|)$. בדוגמה: הרכיבים בגדלים 5, 2, 3 $\Rightarrow$ מוחזר {5, 1}.

שאלה 7

- א. לא נכונה. דוגמה נגדית: הגרף השלם K_4 – עץ DFS שלו הוא מסלול $a-b-c-d$, ואילו כל עץ BFS שלו הוא "כוכב" (השורש מחובר ישירות לשלושת האחרים).
ב. נכונה. בגרף מלא (שלם) כל קודקוד שכן של כל הקודקודים, ולכן DFS אף פעם אינו חוזר אחורה לפני שביקר בכולם: מכל קודקוד הוא ממשיך לקודקוד חדש. עץ ה-DFS הוא מסלול ("שרוך") שעובר בכל הקודקודים – לכל צומת בן אחד בלבד (חוץ מהעלה האחרון).
ג. נכונה. לכל שני קודקודים u, w : יש מסלול $u \rightsquigarrow v$ ומסלול $u \rightsquigarrow w$, ושרשורם הוא מסלול $u \rightsquigarrow w$. לכן בין כל שני קודקודים יש מסלולים בשני הכיוונים.
ד. נכונה. אילו היה מסלול קצר יותר מ- s_m ל- s_k , היינו מחליפים בו את הקטע $s_m \dots s_k$ ומקבלים מסלול מ- s_j ל- s_n קצר יותר מ- S – בסתירה לכך ש- S הקצר ביותר.
ה. לא נכונה. דוגמה נגדית: מעגל באורך 4, $a-b-c-d-a$, הוא גרף דו-צדדי: $\{a, c\}$ ו- $\{b, d\}$.
ו. לא נכונה. דוגמה נגדית: משולש a, b, c שבו משקל כל קשת 1 – כל שתי קשתות הן עץ פורש מינימלי במשקל 2, כלומר יש 3 עצים כאלה.

שאלה 8

- א(1). האלגוריתם של דייקסטרה מ- v_j : אתחול $d[v_j]=0$ ושאר $d[v]=\infty$; בכל איטרציה מעבירים מ- T ל- P את הקודקוד u בעל d מינימלי ומרפים את הקשתות שלו (אם $d[u]+w(u,v) < d[v]$ אז $d[v] \leftarrow d[u]+w(u,v)$, $\pi[v] \leftarrow u$). המסלולים נקראים לאחור לפי π .
א(2). $O(n^2)$ (מימוש במערך). עם ערימת מינימום: $O((n + |E|) \cdot \log n)$.
ב(1). S, G, C, H, B, D, I, F . סדר הקביעה: d/π : S 0/- , G 2/S , C 9/G , H 10/G , B 12/G , D 16/H , I 18/H , F 22/G.
ב(2). קשתות העץ $(\pi[v] - v)$: S-G, G-B, G-C, G-F, G-H, H-D, H-I.

פרק שני – מודלים חישוביים

שאלה 9

א(1). לא נכונה. $L_1 = \{a^n b^n \mid n \geq 0\}$, $L_2 = \{b^n a^n \mid n \geq 0\}$ - שתיהן לא רגולריות, אבל $L_1 \cap L_2 = \{\varepsilon\}$, שפה סופית ולכן רגולרית.

א(2). לא נכונה. $(L_1 \cdot L_2)^2 = \{abab\}$, $L_1 = \{a\}$, $L_2 = \{b\}$, $n = 2$: $L_1^2 \cdot L_2^2 = \{aabb\}$ ואילו $L_1 = \{a\}$, $L_2 = \{b\}$, $n = 2$: $(L_1 \cdot L_2)^2 = \{abab\}$.

ב(1). $aab \notin L$; $\varepsilon \in L$; $baa \in L$; $cbba \in L$ (מכילה ab); $aaccc \notin L$ (5 מופעים של a ו- c אי-זוגי).

ב(2). 6 מצבים (זוגיות של a ו- c האות האחרונה): q_0 (התחלה, זוגי), q_1, q_2, q_3, q_4, q_5 ; מקבלים: q_0 , q_2, q_4 . מעברים: q_3 ; q_1 : $a \rightarrow q_2$, $b \rightarrow q_5$, $c \rightarrow q_0$; q_2 : $a \rightarrow q_3$, $c \rightarrow q_1$; q_3 : $a \rightarrow q_0$, $b \rightarrow q_4$, $c \rightarrow q_1$; q_4 : $a \rightarrow q_3$, $b \rightarrow q_4$; q_5 : $a \rightarrow q_2$, $b \rightarrow q_5$.

$a \rightarrow q_2$, $c \rightarrow q_0$; q_4 : $a \rightarrow q_3$, $b \rightarrow q_4$; q_5 : $a \rightarrow q_2$, $b \rightarrow q_5$ (מעבר חסר = דחייה).

שאלה 10

א(1). $L_5 \cap L_6 = \{110\}$

א(2). $L_2^R = \Sigma^* = L_2$

א(3). $L_1 \cdot L_6 = \emptyset$

א(4). $L_3 \cdot L_4 = \{0110\}$

א(5). $L_4 \cdot L_5 = \{0110, 0110110, 011000, 0110001\}$

ב. דוחפים 3 סימנים לכל ab ו-1 לכל c (הסימן התחתון ביותר מסומן Y), ושולפים סימן אחד לכל b ; מגיעים למצב המקבל q_5 כששולפים את Y . מצבים מקבלים: q_0 (בשביל ε) ו- q_5 . אין מעברי ε .

שאלה 11

א. המכונה עוצרת במצב q_7 והסרט: $\Delta \ \$ \ 1 \ 1 \ \$ \ \Delta \ \vdash \dots$ (בין שני סימני $\$$ - 11, כלומר $2 = 3 - 1$).

ב. $\vdash \Delta \ \$ \ 1 \ \$ \ \Delta \dots$ (המכונה עוצרת ב- q_7).

ג. המכונה מחשבת את ההפרש בערך מוחלט $|x - y|$ של שני המספרים: בסיום על הסרט $x - y$ $\Delta \ \$ \ 1^{|x-y|} \dots$ (התוצאה באונרי בין שני סימני $\$$).

ד. 1. נכון: $\vdash \Delta \ \$ \ 1 \ 1 \ \$ \ \Delta \ \rightarrow \vdash \Delta$. לא נכון: q_6 מגיע לתא $\vdash$ ואין לו מעבר עליו - המכונה נתקעת ולא מגיעה ל- q_7 (וגם חסר ה- $\$$ השמאלי). 3. נכון: $\vdash \Delta \ \$ \ \$ \ \Delta \rightarrow \vdash \Delta$.

פרק שני - תכנות מונחה עצמים

שאלה 12

א(1). $Car, Truck$ ו- $Motorcycle$ יורשות מ- $Vehicle$ (חץ ירושה אל $Vehicle$); $Vehicle$ מכילה עצם מטיפוס $Contract$ (הכלה, המעוין בצד של $Vehicle$).

א(2). $Contract$: name, days, kilo; $Vehicle$: id, contract; Car extends $Vehicle$: seats; $Truck$ extends $Vehicle$: max; $Motorcycle$ extends $Vehicle$: offRoad.

ב(1). $this.contract = new Contract(name, days, kilo); this.id = id$;

ב(2). $Java$: $super(name, days, kilo, id); this.seats = seats; C\#$: $base(name, days, kilo, id) \{ this.seats = seats; \}$

ג(1). $C\#$: $public virtual double Payment() \{ return 60 * this.contract.getDays() + 2 * this.contract.getKilo(); \}$ (עם $GetDays/GetKilo$)

ג(2). Car - אין צורך (יורשת את התעריף הבסיסי). Motorcycle: `return super.payment() + 500;` `base.Payment()` / 2 `(C#: override)`

שאלה 13

AA: `AA() { x = 2 }`, `AA(int x)`, `AA(AA other) { x = other.x }`. BB: `BB() { super(1); f = null }`, `BB(int x, AA f) { super(x); this.f = f }`, `BB(AA f) { super(0); this.f = new AA(f) }`

שאלה 14

x = 4
x = 4 y = 2
x = 2
x = 2 y = 4
x = 1 y = 22
x = 1 y = 22

א.

Temp: x = 1 y = 0
x = 3
x = 4 y = 3
x = 1 y = 10
x = 2 y = 1
x = 1 y = 0
x = 1 y = 0

ב.

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