

# מספר הערות לסטודנטים בעקבות בדיקת מועד ב'-2 בתכנון אלגוריתמים

## שאלה 1

### סעיף א:

טעויות נפוצות:

- במקרים רבים האלגוריתם החדש תואר באופן חסר: "נבצע k חזרות על האלגוריתם הנתון". החלק החשוב ביותר – כיצד מחזירים תשובה לאחר k הרצות – היה חסר או שתואר באופן מרומז בלבד. על כך ירדו עד 2 נקודות.
- חסר הסבר על פולינומיות האלגוריתם החדש.

### סעיף ב:

כיוון שסעיף זה היה ברמת קושי גבוהה יותר משאר הסעיפים, כל מי שהקיף את התשובה הנכונה זכה ללפחות נקודה אחת.

טעויות נפוצות:

- מספר לא מבוטל של תשובות סבבו סביב הרעיון (השגוי) שניתן להתייחס למסלולים בהם הוזרמה הזרימה בעת בנייתה. **הזרימה בשאלה היתה נתונה. לא ניתן לדעת כיצד חושבה. יתרה מכך, קיימת יותר מזרימת מקסימום יחידה עבור רשת זרימה כלשהי ולכן במקרה הכללי לא ניתן להתייחס לאופן חישובה. טענות לגבי קיבולי קשתות או זרימה בקשתות במסלול זה או אחר גם הן צריכות להעשות במשנה זהירות. לדוגמא: לא ניתן לטעון דבר לגבי מסלול הזרימה אליו שייכת צלע מסויימת. היו הסברים מבוססי "דיון מסלולים" כמתואר לעיל שנגעו מעט באינטואיציה נכונה, אך אם צריך היה להרחיבם להוכחה התוצאה היתה ארוכה מאוד ומסורבלת ("הוכחה שאינה סבירה").** תשובות כנ"ל זכו במקרה הטוב לעד 3.5 נקודות.

### סעיף ג:

- מספר סטודנטים מצומצם הוסיפו קיבול 1 רק לקשתות החתך. זוהי דוגמא שגויה ולא ניתנו בזכותה נקודות.

### סעיף ד:

- אין הערות.

## שאלה 2

- 1) Sometimes the formula did not correspond to the definition of opt.
- 2) Often, more or less a verbal description of the formula was given instead of an explanation of the formula correctness.
- 3) Sometimes, the algorithm given in item b did not correspond to the formula given in item a. This is not OK, since we did not ask for an explanation of the algorithm correctness only since the formula's correctness was explained.
- 4) Often, the number of iterations of the algorithm was analyzed only. That is, mentioning the running time  $O(1)$  of an iteration was missing.

5) For a recursive algorithm, not only computations take time, but also the recursive calls themselves. That is, a recursive call takes time even if there is no computation in it. This was ignored sometimes.

6) In algorithm analysis, any time bound always should be expressed in the terms of the given parameters. In our case, in terms of  $n$ . Linearity in the output size asked in item 3 is an ADDITIONAL requirement to the time analysis. Often in item c, no legal time bound as above was given, only a bound in the terms of the output size.

Most of the above errors (except of 2 and 5) caused only a very small point reduction.

### שאלה 3

#### סעיף א – טעויות נפוצות

- דרישה לעד שהוא קליקה בגודל  $\sqrt{|V|} \geq$  או  $\sqrt{|V|} \leq$  (ללא התייחסות ל- $k$ ).
- חסרה בדיקה כי  $k \leq \sqrt{|V|}$ .

#### סעיף ב

מפתח ניקוד: הקפת התשובה הנכונה – 1, רדוקציה – 6, הוכחת נכונות – 6, זמן ריצה – 2 – טעויות נפוצות

- הגדרת הרדוקציה לפי ממיר קלט, קופסה שחורה, וממיר פלט.
- שימוש באלגוריתם שפותר את *Clique* או  $\sqrt{Clique}$ .
- הוספת כמות קודקודים לא נכונה.
- לא רושמים מפורשות מה הרדוקציה מחזירה כפלט.
- שימוש בעד של אלגוריתם האימות בשלב הגדרת הרדוקציה או ההוכחה.
- זמן ריצה של  $O(k)$  ללא טיפול במקרה  $k > |V|$  (ייתכן זמן ריצה שאינו פולינומי).
- הוספת קליקה לגרף המקורי.
- פונקציית הרדוקציה משנה רק את  $k$  (ולא את הגרף המקורי).
- התייחסות ל- $k, \sqrt{|V|}$  של הקלט, ולא של פלט הרדוקציה, בשלב ההוכחה.
- ייצור גרף בו קיימת קליקה בגודל בדיוק  $\sqrt{|V|}$  (זו אינה טעות לכשעצמה, אך זה מיותר, ורוב הסטודנטים שנקטו בגישה זו טעו בחישוב כמות הקודקודים שיש להוסיף).
- הנחה מוטעית כי  $\binom{|V|}{\sqrt{|V|}}$  הינו מספר פולינומי ב- $|V|$ .

### שאלה 4

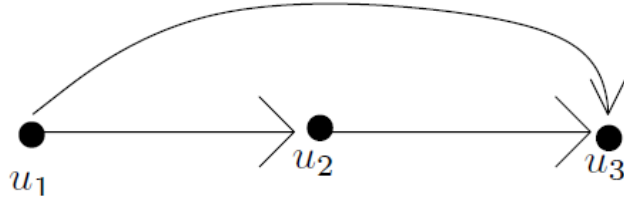
מפתח ניקוד:

סעיף א': אלגוריתם 6 נק'; הוכחה 5 נק'; זמן ריצה 2 נק'  
 סעיף ב': אלגוריתם 10 נק'; זמן ריצה 2 נק'

- מי שעשה אלגוריתם בזמן ריצה  $|E| \cdot |V|$ , קיבל הורדה של 3 נקודות.
- פתרון נכון שלא מציין איך מוצאים מסלול מ- $u_1$  ל- $u_{i+1}$  (יש דרכים לא נכונות לעשות את זה) – 3 נק'
- הוכחה שכוללת מקרים שלא משלימים אחד את השני. למשל "כיוון 1: קיים מסלול שכולל את כל איברי  $U$ " ואחרי זה "כיוון 2: האלגוריתם החזיר לא". אי אפשר לערבב את שתי הדרכים. הורדה של 4-5 נקודות (תלוי כמה טוב כל חלק הוכח בפני עצמו).

## דרכים שגויות לפתרון

- אי אפשר להניח שהגרף קשיר.
- מי שנתן פתרון שעובר על כל המסלולים האפשריים, בעיקרון צדק, אבל קיימת כמות אקספוננציאלית של מסלולים, ולכן פתרון זה לא קיבל ניקוד. הסבר בקצרה שזו כמות אקספוננציאלית: נחשוב על הקודקודים בסדר של המיין הטופולוגי, ונניח שלכל קודקוד יש צלע יוצאת לכל מי שבא אחריו במיין. מסלול מתחיל בקודקוד הראשון, ויכול להגיע לקודקוד שאחריו במיין או לא. ואותו דבר לקודקוד הבא. באופן כללי לכל קודקוד (מלבד הראשון) יש אפשרות לבחור להיות חלק מהמסלול או לא להיות חלק מהמסלול באופן בלתי תלוי באחרים. שזה כמו מילה בינארית.
- מי שעשה DFS מ- $u_1$ , ולא מתחשב באפשרות ש-DFS יקבל בחירות אקראיות. למשל בדוגמה הבאה DFS יכול לבחור בקשת  $(u_1, u_3)$ , ולסיים את הריצה שלו.



- הגרף המושרה מקודקודי  $U$  יכול להיות גרף ללא צלעות. למשל

$$E = \phi$$

$$V = \{v_1, u_1, u_2\}$$

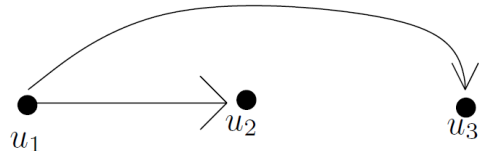
$$U = \{u_1, u_2\}$$

$$E = \{(u_1, v_1), (v_1, u_2)\}$$



- פתרונות מהסוג:

- (1) מיין טופולוגית.
  - (2) מצא  $u_1$  הראשון במיין ששייך ל- $U$ .
  - (3) תחפש את כל מי שנגיש מ- $u_1$ , ובדוק אם הגענו לכל איברי  $U$ .
- פתרון זה לא עובד כי יתכן המצב הבא שבו כולן נגישים, אבל אין מסלול בין כל איברי  $U$ .



כמו כן, שימו לב שהדוגמאות פשוטות, אך כל קודקוד ניתן להחלפה בתת גרף גדול כרצוננו, והאלגוריתמים יטעו על כל התת גרף המדובר, ולא מדובר בשגיאה של קודקוד בודד שניתן לטפל בו עם בדיקת if פשוטה.

- **סעיף ב:** מי שלא התייחס לזה שצריך להמיר את  $U$  בהתאם לקודקודים החדשים, 5 נק'.