מועד א', סמסטר ב', תש"ף (2019-2020) 17.07.20

<u>מבחן ב"אלגוריתמים"</u>

(מרצים: חיים קפלן ואורי צוויק)

(מתרגלים: טל ינקוביץ וג'אד סילבק)

משך המבחן : 3.5 שעות (כולל זמן התארגנות וסריקה)

המבחן עם חומר **סגור**, פרט ל"דף נוסחאות" שפורסם מראש.

ענה/י על ארבעת השאלות הבאות. משקל כל השאלות שווה.

<u>שאלה 1 (25 נקודות)</u>

בשאלה הזאת k הוא מספר טבעי. לא ניתן להניח שהוא קבוע אבל ניתן להניח שהוא קטן ביחס ל-$|V|$, מספר הצמתים ברשת שבה אנחנו דנים.

א. נתונה רשת זרימה $G = (V, E, c, s, t)$ כאשר $c: E \rightarrow \{1,2, \dots, k\}$, כלומר כל קיבולי הקשתות הם מהקבוצה $\{1,2, \dots, k\}$. אין ברשת קשתות מקבילות. תן/י חסם (אסימפטוטי) טוב ככל האפשר על זמן הריצה של האלגוריתם של Ford-Fulkerson על רשת זו. החסם צריך להיות במונחים של $|V|, |E|$ ו-k. הסבר/י את תשובתך.

ב. נתונה רשת זרימה $G = (V, E, c, s, t)$. נתון שלכל צומת ב-G, פרט למקור ולבור, יש רק קשת נכנסת אחת, והיא בקיבול k, או רק קשת יוצאת אחת, והיא בקיבול k. בנוסף לכך, נתון שקיבולי כל שאר הקשתות הם **לפחות** k. אין ברשת קשתות מקבילות. תאר/י אלגוריתם יעיל ככל האפשר למציאת זרימת מקסימום ברשת. מה סיבוכיות האלגוריתם? הסבר/י את תשובתך.

ג. נתונה רשת זרימה $G = (V, E, c, s, t)$. נתון שלכל צומת ב-G, פרט למקור ולבור, יש רק קשת נכנסת אחת, והיא בקיבול k, או רק קשת יוצאת אחת, והיא בקיבול k. בנוסף לכך, נתון שקיבולי כל שאר הקשתות הם **לכל היותר** k. אין ברשת קשתות מקבילות. תאר/י אלגוריתם יעיל ככל האפשר למציאת זרימת מקסימום ברשת. מה סיבוכיות האלגוריתם? הסבר/י את תשובתך.

שאלה 2 (25 נקודות)

נתון גרף מכוון $G = (V, E)$ עם פונקציית **מחיר** אי-שלילית $c: E \to \mathbb{R}^+$ על קשתותיו, ופונקציית **רווח** אי-שלילית $p: V \to \mathbb{R}^+$ על צמתיו. **העלות** $a(P)$ של מסלול $P = (u_0, u_1, \dots, u_k)$ בגרף מוגדרת להיות $a(P) = \sum_{i=1}^{k-1} c(u_i, u_{i+1}) - \sum_{i=1}^{k} p(u_i)$. ניתן להניח שהגרף G הוא קשיר בחוזקה.

א. תאר/י אלגוריתם יעיל ככל האפשר שבהינתן שני צמתים s ו-t בגרף מוצא מסלול בעל עלות מינימאלית מ-s ל-t בגרף או אומר שאין מסלול מ-s ל-t בעלות מינימלית. (הסבר/י מתי מצב כזה יכול לקרות.) מה סיבוכיות האלגוריתם? הסבר/י מדוע האלגוריתם נכון.

ב. נתון עתה שעבור כל צומת v בגרף, העלות המינימלית של מסלול מ-s ל-v מוגדרת היטב ושווה ל-$f(v)$. תאר/י אלגוריתם יעיל ככל האפשר שבהינתן הערכים $f(v)$, עבור כל $v \in V$, מוצא עבור כל צומת v את העלות המינימלית של מסלול מ-v ל-t. מה סיבוכיות האלגוריתם? הסבר/י מדוע האלגוריתם נכון.

שאלה 3 (25 נקודות)

נתונה התוכנית הליניארית הבאה P .

$$
(P) \qquad
\begin{array}{ll}
\max & c_1x_1 + c_2x_2 \\
s.t. & a_1x_1 + b_1x_2 \le d_1 \\
 & a_2x_1 + b_2x_2 \le d_2 \\
 & \vdots \\
 & a_mx_1 + b_1x_m \le d_m \\
 & x_1, x_2 \ge 0
\end{array}
$$

התוכנית היא בצורה סטנדרטית. בתוכנית שני משתנים ו-m אילוצים. נתון שהתוכנית P היא חסומה ופיזיבילית.

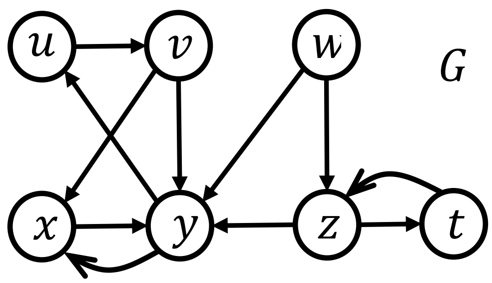
א. תאר/י אלגוריתם שסיבוכיותו $O(m^2)$ למציאת פתרון אופטימלי של P. הסבר/י את תשובתך.

ב. כתוב/י את התוכנית הדואלית לתוכנית P.

ג. תאר/י אלגוריתם יעיל ככל האפשר למציאת פתרון אופטימלי של התוכנית הדואלית. (על האלגוריתם להחזיר פתרון אופטימלי לתוכנית הדואלית, לא רק את ערכו.) מה סיבוכיות האלגוריתם, כפונקציה של m? הסבר/י את תשובתך.

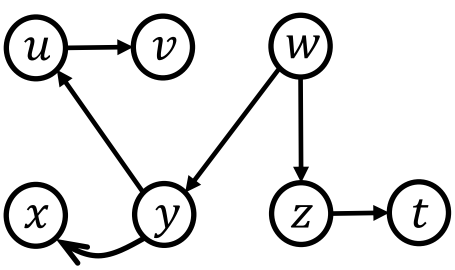
שאלה 4 (25 נקודות)

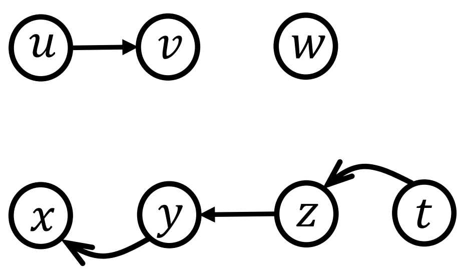
נתון הגרף המכוון הבא:

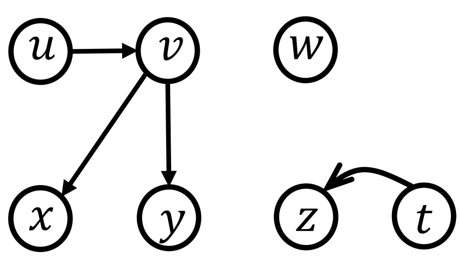


א. עבור כל אחד משלושת מהיערות הבאים,
ציין/י האם הוא יכול להתקבל כפלט של DFS על הגרף G?
אם היער אינו יכול להתקבל. הסבר/י מדוע. אם היער יכול להתקבל, סמן/י זמני גילוי וסיום אפשריים של הצמתים, וסווג/י את קשתות הגרף לפי ארבעת הסוגים האפשריים: tree, forward, backward, cross.

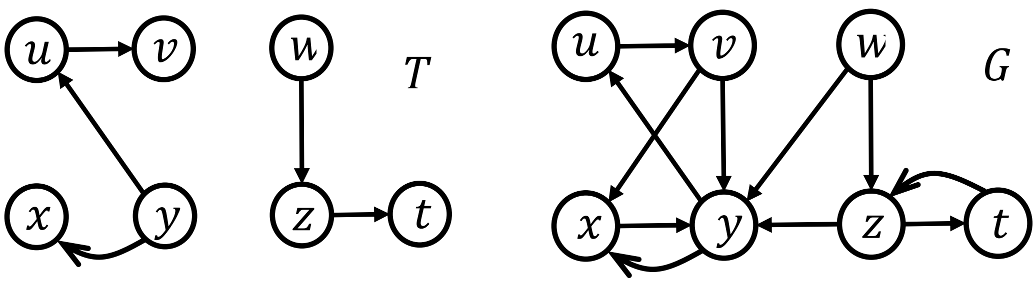
העתק/י את כל אחד מהיערות לדפי התשובה שלך.







ב. נריץ את אלגוריתם ה-DFS הכפול (האלגוריתם של Sharir-Kosaraju) למציאת רכיבי קשירות חזקה ב-G. נניח כי T (בציור למטה) הוא היער שהתקבל כתוצאה מסריקת ה-DFS הראשונה. כתוב/י זמני גילוי וסיום אפשריים של כל הצמתים, תאר/י את היער שיתקבל בסריקת ה-DFS השניה על הגרף ההפוך. באיזה סדר יתגלו רכיבי הקשירות החזקה?



ג. נריץ עתה את האלגוריתם של Tarjan על G למציאת רכיבי קשירות חזקה. שוב נניח כי עץ ה-DFS שמחשב האלגוריתם הוא T. תאר/י את ההשמות לערכי ה-low של הצמתים לפי הסדר (היצמדו לזמני הגילוי והסיום מסעיף ב). מתי במהלך הריצה מזהה האלגוריתם את כל אחד מרכיבי הקשירות החזקה? (לנוחיותכם מצורף בעמוד הבא הפסאודו-קוד של האלגוריתם שנלמד בכיתה.)

בהצלחה !!!

Tarjan's SCC algorithm

```
SCC(G)
For each vertex $u \in G.V$
    u.color = WHITE
    u.$\pi$ = NIL
time = 0
For each vertex $u \in G.V$
    if u.color == WHITE
        SCC-Visit(G,u)
```

```
SCC-Visit(G,u)
time = time + 1
u.d = low(u) = time
push(u,S)
u.color = GRAY
For each vertex $v \in G.Adj[u]$
    if v.color == WHITE
        v.$\pi$ = u
        SCC-Visit(G,v)
        low(u) = $\min\{low(u), low(v)\}$
        else if $v \in S$
                low(u) = $\min\{v.d, low(u)\}$
     u.color = BLACK
     time = time + 1
     u.f = time
     if low(u) = u.d then perform pop(S) until u is popped,
     the set popped is a SC
```