ממן 11

שאלה 1

סעיף א

עבור כל לוח זמנים, עלינו לבחור נמל שבו הספינה תעצור עד סוף החודש. באופן זה, יוגדרו למעשה הקיצוצים בלוחות הזמנים. נגדיר בעיית זיווג יציב עם ספינות ונמלי-ים. כל ספינה תדרג רשימת העדפות של נמלי-ים, בסדר הכרונולוגי שבו היא מבקרת בהם. כל נמל-ים ידרג רשימת העדפות של ספינות בסדר כרונולוגי-הפוך מסדר ביקור הספינות בנמל-הים.

<u>טענה</u>: זיווג יציב בין ספינות לנמלי-ים מגדיר קיצוץ חוקי של לוחות הזמנים (כך שלוחות הזמנים עומדים בתנאי ◇ המוצג בשאלה).

<u>הוכחה</u>: יהי זיווג יציב בין ספינות לנמלי ים המבוסס על רשימות ההעדפות שהגדרנו לעיל (אשר נוצר ע"י אלגוריתם G-S), ונניח בשלילה כי לוחות הזמנים לא עומדים בתנאי ◇ כלומר, ספינה S_i כלשהי נכנסת לנמל P_k זמן שספינה S_j ($j\neq i$) כבר עצרה שם (ונמצאת שם עד סוף החודש). אולם, ממצב זה נובע על פי רשימות ההעדפות שהגדרנו לעיל כי הספינה S_i מעדיפה את נמל P_k על הנמל שבו היא אמורה לעצור עד סוף החודש והנמל P_k מעדיף את הספינה S_i על הספינה S_j. הדבר סותר את ההנחה שלנו שהזיווג שנוצר בהתבסס על רשימות ההעדפות שהגדרנו הוא יציב. מכאן שלוחות הזמנים עומדים בתנאי ◇.

הראינו אפוא כיצד ניתן תמיד למצוא קיצוץ ללוחות הזמנים, אשר עומד בתנאי ◇)והאלגוריתם הבונה אותם הוא אלגוריתם G-S).

סעיף ב

נביא דוגמא לקבוצה של רשימות העדפות עבורה קיימת החלפה שתקדם את בן הזוג של אישה שהחליפה העדפות (כלומר שיקרה ברשימת ההעדפות שלה):

נתבונן בבעיית הזיווג היציב כאשר n=3 ורשימות ההעדפות הן כדלקמן:

m_1	m_2	m_3	w_1	w_2	w_3
w_3	w_1	w_3	m_1	m_1	m_2
w_1	w_3	w_1	m_2	m_2	m_1
w_2	w_2	w_2	m_3	m_3	m_3

לאחר הרצת אלגוריתם G-S נקבל את הזיווגים הבאים:

$m_1 – w_3$,
$m_2 – w_1$,
$m_3 – w_2$.

נשים לב כי w_3 קיבלה את m_1, למרות שהיא הייתה מעדיפה את m_2. כעת נראה שאם w_3 תשקר ברשימת ההעדפות שלה היא תוכל לקבל m_2 במקום m_1, כפי שהיא מעדיפה.

נניח כעת שרשימות ההעדפות של כל הגברים ושל הנשים w_1 ו- w_2 נשארות זהות אך w_3 מציגה את רשימת ההעדפות השקרית הבאה:

w_3'
m_2
m_3
m_1

כעת, על ידי הרצת אלגוריתם G-S נקבל את הזיווגים הבאים:

$m_1 - w_1$,
$m_2 - w_3$,
$m_3 - w_2$.

קל לראות כי הפעם w_3 קיבלה את m_2 אשר מהווה את העדפתה האמתית הראשונה. מכאן שרשימת העדפות שקרית של אישה כלשהי יכולה לשפר את הזיווג הסופי שהיא מקבלת באלגוריתם G-S.

שאלה 2

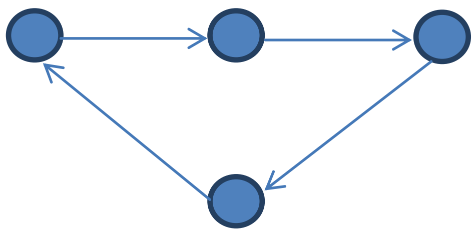
סעיף א

יהי G = (V, E) גרף קשיר בעל לפחות שני קדקודים. אז מתקיימת אחת מבין האופציות הבאות:

(1) אם הגרף מכיל צומת v שדרגתו היא 1, אזי היא מחוברת בקשת רק לצומת אחד ולכן ברור כי ניתן להוריד את הצומת v מבלי לפגוע בקשירות הגרף.
(2) אחרת, כל הצמתים בגרף בעלי דרגה השווה ל- 2 לפחות. נחפש את הגשרים בגרף לפי אלגוריתם הגשרים (עמוד 37 במדריך הלמידה). נמצא בגרף צומת u שלא יוצא ממנו קשת שמהווה את אחד הגשרים שנמצאו – צומת כזה קיים מאחר ולפי ההנחה, כל הצמתים בגרף בעלי דרגה השווה ל- 2 לפחות, ולכן הגרף מכיל מעגל, ומכאן לפי טענה 3.13 במדריך הלמידה (עמוד 33), חייבת להיות בגרף קשת שאינה גשר. לסיום, צומת u הוא הצומת שאותו נסיר מבלי לפגוע בקשירות הגרף.

הראינו אפוא שבכל מקרה, עבור Gקשיר בעל לפחות שני קדקודים, קיים צומת ב- G שלאחר הסרתו עדיין G יישאר קשיר.

סעיף ב



קל להיווכח כי בין <u>כל</u> שני צמתים v_1 ו- v_2 בגרף המכוון שלעיל קיים מסלול מ- v_1 ל- v_2 ומסלול מ- v_2 ל- v_1 ולכן הגרף קשיר היטב. אולם לאחר הסרה של <u>כל</u> צומת מן הגרף, תמיד יתקבל גרף שאינו קשיר היטב.

שאלה 3

ראשית, נוכיח מספר טענות:

<u>טענה 1:</u> בגרף לא מכוון קשיר, אם אין אף מעגל, אזי לא ניתן לכוון את הקשתות כך שלכל צומת תהיה דרגת כניסה גבוהה מאפס.
<u>הוכחה:</u> יהי G = (V, E) גרף לא מכוון קשיר ללא מעגלים. נניח בשלילה שקיימת דרך לכוון את קשתות G כך שנקבל לכל צומת דרגת כניסה גבוהה מאפס. נבצע את הכיוון הנ"ל ונקבל גרף מכוון ללא מעגלים. אולם, לפי טענה (3.19) בעמוד 110 בספר הלימוד קיים צומת $v \in V$, שלא נכנסת אליו אף קשת. לכן דרגת הכניסה של v היא 0 וקיבלנו סתירה.

<u>טענה 2:</u> בגרף לא מכוון קשיר, מספיק שיהיה מעגל אחד על מנת שניתן יהיה לכוון את הקשתות כך שלכל צומת תהיה דרגת כניסה גדולה מאפס.
<u>הוכחה:</u> יהי G = (V, E) גרף לא מכוון קשיר המכיל בדיוק מעגל אחד. כעת, נראה כיצד ניתן לכוון את הקשתות כך שלכל צומת בגרף תהיה דרגת כניסה גדולה מאפס. ראשית, נבחר באחת מקשתות המעגל. תהי קשת זאת $e = \{v_1, v_2\}$ וניתן לה כיוון כרצוננו, או מ- v_1 ל- v_2 או הפוך. נניח בלי הגבלת הכלליות שהחלטנו לכוון את הקשת מ- v_1 ל- v_2. כעת נריץ אלגוריתם BFS כאשר השורש הוא v_2 ונכוון את הקשתות (מלבד אלו שכבר כוונו) כך שהן ייצאו מצמתים בשכבות בעלות מספר סידורי נמוך לכיוון צמתים בשכבות בעלות מספר סידורי גבוה (למשל מצומת בשכבה L_0 לצומת בשכבה L_1). קיבלנו ש- v_2 הוא שורש של עץ BFS של הגרף הקשירG ולכן באמצעות ההכוונה שביצענו, ניתן להגיע מ- v_2 לכל צומת אחר של G. לכל צומת בעץ יש אב, מלבד ל- v_2 מכיוון שהוא השורש, ולכן לכל צומת פרט ל- v_2 נכנסת קשת ומכאן שדרגת הכניסה של כל צומת פרט ל- v_2 גדולה מ- 0. אולם גם ל- v_2 נכנסת קשת– זוהי הקשת מ- v_1 אשר כיוונו טרם ביצוע ה- BFS, ולכן גם דרגת הכניסה של v_2 גדולה מ- 0. על כן, ביצענו הכוונה של קשתות G כך שדרגת הכניסה של כל צמתי G גדולה מ- 0.

כעת על סמך שתי הטענות שלעיל, נוכל לבנות את האלגוריתם המבוקש:

1. מצא את כל רכיבי הקשירות של G.
2. עבור כל רכיב קשירות'G אשר מוכל או שווה ל- G:
   1.2. אם $|E_{G'}| = |V_{G'}| - 1$, אזי רכיב הקשירות אינו מכיל מעגל (לפי טענה 3.2 בעמוד 84 בספר) **ולכן** החזר: "לא ניתן לכוון את הקשתות כמבוקש" (לפי טענה 1).
   2.2. אחרת, כוון את הצמתים כפי שמתואר בהוכחת טענה 2.
3. החזר את כיווני הקשתות.

נכונות האלגוריתם נובעת באופן מידי מטענות 1 ו- 2 שהוכחו לעיל, תוך כדי שימוש בהן על כל רכיבי הקשירות של G על מנת להחזיר הכוונה כנדרש עבור <u>כל</u> הקשתות של G.

סיבוכיות זמן הריצה של האלגוריתם: מציאת כל רכיבי הקשירות מתבצעת בסיבוכיות $O(m+n)$ (למשל ע"י הרצת האלגוריתם DFS-Loop שבעמוד 27 במדריך הלמידה) ולכן זוהי סיבוכיות סעיף 1. סיבוכיות סעיף 2.1 בוודאי גם היא חסומה מלמעלה ע"י סדר גודל שלm+n, כלומר סעיף 2.1 מתבצע ב- $O(m+n)$ ולא יותר. סעיף 2.2 דורש למצוא מעגלולבצע את אלגוריתם BFS עם תוספת של פעולות בסיבוכיות $O(1)$ עבור כל צומת. מציאת מעגל יכולה להתבצע בסיבוכיות $O(m+n)$ למשל ע"י הרצת DFS וחיפוש קשתות אחורה. אלגוריתם BFS עם תוספת של פעולות בסיבוכיות $O(1)$ עבור כל צומת מניב סיבוכיות $O(m+n)$ ולכן בסה"כ סעיף 2.2 מתבצע בסיבוכיות $O(m+n)$. סעיף 3 מתבצע בסיבוכיות $O(1)$.**מכאן, שהאלגוריתם כולו מתבצע בסיבוכיות זמן ריצה של $O(m+n)$.**

שאלה 4

יהי גרף G = (V, E) וצמד צמתים $s, t \in V$. אנו מעוניינים למצוא את המסלול בין s ל- t שבו סכום הדרגות של הצמתים לאורך המסלול הוא מזערי. לצורך כך, ראשית נעבור על רשימת הסמיכויות ועבור כל צומת $u \in V$ נשמור שדה *d(u)* השווה לדרגת הצומת *u*. כעת, נעבור על כל צמתי הגרף ועבור <u>כל</u> צומת $v \in V$ של הגרף נוסיף לגרף "שרוך" של צמתי סרק באורך *d(v)* אשר שורשו הוא *v* ואת הצומת האחרון של "השרוך" נחבר בקשתות לכל שכניו המקוריים של *v*. כלומר, נחבר את *v* בקשת לצומת סרק חדשv_1, את הצומת v_1נחבר בקשת לצומת סרק חדש v_2, ..., את הצומת $v_{d(v)-1}$ נחבר בקשת לצומת סרק חדש $v_{d(v)}$, ולבסוף את הצומת $v_{d(v)}$ נחבר בקשתות לכל שכניו המקוריים של הצומת *v*. כלומר, הארכנו את המסלול בין כל שני צמתים שכנים על פי דרגת הצמתים. כעת, נריץ את אלגוריתם *BFS* על הצומת *s* ונקבל עץ- *BFS* מושרש ב- *s*, שנסמנו ב- *T*. כעת נתחיל בצומת *t*בעץ *T*ונעלה באופן רקורסיבי עד צומת *s* באמצעות שדות ה- *parent* של כל צומת (אשר בהם שמרנו מצביע לצומת האב בעץ ה- *BFS* תוך כדי יצירתו). תוך כדי מעבר זה על המסלול בין *t* ל- *s*, נשמור את המסלול אך ללא צמתי הסרק וקשתות הסרק שנוצרו ביחיד עם צמתי הסרק. לבסוף נחזיר את המסלול ששמרנו בסדר הפוך.

כלומר האלגוריתם בפסאודו-קוד הוא:

1. נעבור על כל צמתי הגרף ועבור כל צומת $u \in V$ של הגרף נשמור שדה *d(u)* המכיל את דרגת הצומת (נעשה זאת באמצעות רשימת הסמיכויות).
2. נשנה את הגרף *G* באופן הבא: עבור כל צומת $v \in V$ של הגרף, נחבר את *v* בקשת לצומת סרק חדש v_1, את הצומת v_1 נחבר בקשת לצומת סרק חדש v_2, ..., את הצומת $v_{d(v)-1}$ נחבר בקשת לצומת סרק חדש $v_{d(v)}$, ולבסוף את הצומת $v_{d(v)}$ נחבר בקשתות לכל שכניו המקוריים של הצומת *v*. נקרא לגרף שיצרנו *G'*.
3. נריץ *BFS(s)* על הגרף *G'*, כלומר אלגוריתם *BFS* כאשר הצומת בשכבה L_0 הוא *s*. נסמן את עץ ה- *BFS* שקיבלנו כ- *T*.
4. נבצע בעץ *T* את הפעולות הבאות: נתחיל בצומת *t* ונחזור עד *s* באמצעות שדות ה- *parent* של כל צומת (החל ב- *t*), ותוך כדי המעבר נשמור את המסלול אך ללא צמתי הסרק וקשתות הסרק (שנוצרו עם צמתי הסרק בסעיף 2 של האלגוריתם).
5. נחזיר את המסלול ששמרנו בסדר הפוך.

<u>נכונות האלגוריתם:</u> על פי עמוד 26 במדריך הלמידה, המסלול בין שני צמתים בעץ *BFS* הוא מהסלול הקצר ביותר בין שני צמתים אלו בגרף המקורי שממנו נוצר עץ ה- *BFS* על ידי הרצת אלגוריתם *BFS* (הדבר נובע מטענה 3.3 בעמוד 86 בספר הלימוד). בסעיף 2, יצרנו מהגרף *G* גרף אשר מתחשב בדרגות הצמתים של הגרף המקורי. כלומר, קל לראות כי בתום סעיף 2, אנו מקבלים גרף *G'* כך שבהינתן שני צמתים *s* ו- *t* השייכים לגרף המקורי *G* ובהינתן שני מסלולים R_1 ו- R_2 בגרף *G'* בין *s* ו- *t*, אם בלי הגבלת הכלליות R_1 ארוך מ- R_2 אזי R_1 נוצר ממסלול בין *s* ל- *t* ב- *G* שבו סכום הדרגות של הצמתים גדול יותר מהמסלול האנלוגי שממנו נוצר R_2 (גם הוא מסלול בין *s* ל- *t* ב- *G*). לכן, ככל שסכום הדרגות של הצמתים במסלול כלשהו בין *s* ל- *t* בגרף *G* גדול יותר, כך המסלול הנוצר בגרף *G'* מהמסלול המקורי יהיה ארוך יותר. כמו כן, אלגוריתם *BFS* "מעדיף" את המסלולים הקצרים ביותר, כלומר לכל $j \geq 1$, השכבה L_j שמתקבלת מסריקת *BFS*, כוללת את כל הצמתים הנמצאים <u>בדיוק</u> במרחק *j* משורש עץ ה- *BFS* (טענה 3.3 בעמוד 86 בספר הלימוד). לכן, המסלול בין *s* ל- *t* בעץ ה- *BFS* (דהיינו *T*) יהיה המסלול שנוצר מהמסלול בין *s* ל- *t* ב- *G* אשר בו סכום הדרגות של הקדקודים הוא מזערי. לכן, לאחר שנוריד בסעיף 4 את צמתי הסרק וקשתות הסרק, נקבל את המסלול בין t ל- s ב- G שבו סכום הדרגות של הצמתים לאורך המסלול הוא מזערי. לאחר שבסעיף 5 נהפוך אץ המסלול, נקבל את המסלול בין s ל- t ב- G שבו סכום הדרגות של הצמתים לאורך המסלול הוא מזערי, כנדרש.

<u>סיבוכיות זמן ריצה של האלגוריתם:</u> סעיף 1 דורש מעבר על רשימת הסמיכויות של כל צומת ולשמור שדה *d* בכל צומת. המעבר מתבצע בסיבוכיות $O(\sum_{v \in V} d(v)) = O(m)$ (לפי טענה (3.9) בעמוד 95 בספר הלימוד) ושמירת שדה *d* עבור צומת יחיד מתבצעת ב- *O(1)*. לכן סיבוכיות סעיף 1 היא *O(m)*. סעיף 2 דורש לעבור על כל צמתי הגרף ועבור כל צומת *v* לבצע עבודה בסיבוכיות *O(d(v))* (הוספת *d(v)* צמתים ו- $d(v) - 1$ קשתות). לכן סיבוכיות הסעיף גם היא $O(\sum_{v \in V} d(v)) = O(m)$. סיבוכיות סעיף 3, אשר בו מתבצע אלגוריתם *BFS*, היא $O(m + n)$ (טענה 3.11 בעמוד 98 בספר הלימוד). סעיף 4 דורש מעבר על מסלול בין שני צמתים בעץ וברור כי הדבר מתבצע ע"י פחות פעולות ממעבר על כל קשתות וצמתי העץ ומכאן ברור שהסעיף מתבצע ב- $O(m + n)$ (החסם לא הדוק). סעיף 5 מתבצע ב- *O(1)*. **סה"כ האלגוריתם מתבצע בזמן ריצה של $O(m + n)$.**