<u>תכנון אלגוריתמים תרגיל 1 – דף תשובות</u>

אנא הגישו רק חלק זה. <u>אל תחרגו מהמקום המוקצה לתשובה!</u>

שאלה 1 סעיף א' – תיאור הרדוקציה

בהינתן מופע $G=(V,E)$ ושני קודקודים שונים $s,t\in V$ לבעיית המסלול הקצר, נוסיף צלע חדשה $e=(s,s)$, מקודקוד s לעצמו. נקבל גרף G' שמהווה מופע לבעיית המסלול הלא פשוט. נפעיל את "הקופסא השחורה" (הפותרת את בעיית המסלול הלא פשוט) על G' ונקבל תשובה d, המהווה את אורך המסלול הלא פשוט הקצר ביותר בין s ל-t ב- G'.

הפתרון לבעיית המסלול הקצר על מופע G הוא $d-1$.

שאלה 1 סעיף א' – הוכחת נכונות

<u>רעיון ההוכחה:</u> נוסיף צלע מקודקוד ההתחלה במסלול לעצמו, וע"י כך ניצור מעגל בגודל 1 במסלול. מכך שמעגל זה הוא בגודל מינימלי, בהכרח יבחר מעגל זה או מעגל באורך זהה לו במהלך ריצת האלגוריתם לפתרון בעיית המסלול הלא פשוט, ובכך נדע בדיוק מהו אורך המסלול הקצר ביותר בין קודקודים אלה (הבעיה התחילה מכך שלא ידענו מה אורך המעגל שימצא במסלול).

<u>טענה ראשית:</u> אורך המסלול הקצר ביותר בין שני קודקודים s ו-t הוא 1 פחות מאורך המסלול הלא פשוט הקצר ביותר בגרף בו קיימת צלע מאחד הקודקודים s או t לעצמו.

<u>טענת עזר:</u> אם קיימת צלע מקודקוד s או t לעצמו, בהכרח המסלול הלא פשוט הקצר ביותר ביניהם יכיל מעגל באורך אחד.

<u>אבחנה:</u> מעגל באורך 1 בגרף מכוון הוא מעגל מינימלי, מכיוון שלא ניתן לקצר אותו.

<u>הוכחת הטענה הראשית:</u> בהינתן מופע G לבעיית המסלול הקצר ופתרון d (שקיבלנו מהאלגוריתם של הרדוקציה), נשים לב כי בעת תרגום הקלט הוספנו צלע מקודקוד s לעצמו (במידה ולא הייתה קיימת), כך שיצרנו מעגל באורך 1 על המסלול s..t. מטענת העזר נובע כי פתרון d שנקבל יהיה אורך המסלול הקצר ביותר בין s ל-t המכיל מעגל, ומעגל זה יהיה באורך 1. מכך שבחרנו ליצור את המעגל כבר בקודקוד s, מובטח לנו שאחד המסלולים הקצרים ביותר (הפשוטים) בין s ו-t בהכרח יבחר ע"י אלגוריתם למציאת

המסלול הלא פשוט הקצר ביותר, אחרת ניתן היה לקצר את המסלול הלא פשוט ע"י בחירת המעגל באורך 1 שיצרנו ואת המסלול הקצר ביותר בין s ל-t.

הוכחת טענת העזר: נניח בה"כ כי קיימת צלע מקודקוד s לעצמו, כלומר מעגל באורך 1 המתחיל ומסתיים ב-s. לפי האבחנה ובאופן טריוויאלי, מעגל באורך 1 הוא מעגל מינימלי. נניח בשלילה שהמסלול הלא פשוט הקצר ביותר בין s ו-t מכיל מעגל באורך גדול מ-1 – ניתן היה להחליף מעגל זה במעגל באורך 1 מ-s לעצמו, ובכך לסתור את המינימליות של המסלול הלא פשוט הקצר ביותר, ולכן מסלול זה בהכרח יכיל מעגל באורך 1 (אך לאו דווקא את המעגל שמתחיל ומסתיים ב-s).

שאלה 1 סעיף א' – ניתוח זמן ריצה

המרת קלט – הוספת צלע – $O(|V|)$ – נחפש את s ונוסיף צלע בינה לבין עצמה.

זמן ריצת הקופסא השחורה – $O(|V|+|E|)$

המרת פלט – להחסיר 1 – $O(1)$

סך הכל: $O(|V|+|E|)$

שאלה 1 סעיף ב – תיאור הרדוקציה

שאלה 1 סעיף ב' – הוכחת נכונות

שאלה 1 סעיף ב' – ניתוח זמן ריצה

שאלה 2 סעיף א' – תיאור הרדוקציה מבעיה א' לב'

שאלה 2 סעיף א'– הוכחת נכונות

שאלה 2 סעיף א' – ניתוח זמן ריצה

שאלה 2 סעיף א' – תיאור הרדוקציה מבעיה ב' לא'

שאלה 2 סעיף א'– הוכחת נכונות

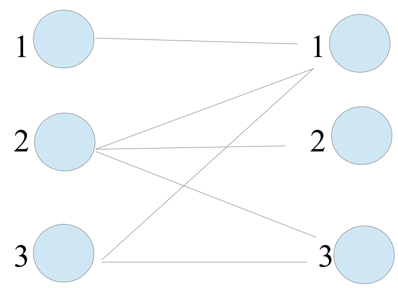
שאלה 2 סעיף א' – ניתוח זמן ריצה

שאלה 2 סעיף ב'

שאלה 2 סעיף ג'

בנחה שקיים אלגוריתם שפותר את בעיה א' בסדר גודל זמן ריצה של $O(n+m)$, עדיין סביר שקיים אלגוריתם (לא מבוסס רדוקציה), שפותר את בעיה ב' באותו הזמן. <u>בהנחה</u> שבמקומות שונים בקוד של האלגוריתם שפותר את בעיה א' מתבצעות בדיקות האם צלעות מסויימות קיימות או לא קיימות בגרף, ניתן לומר שאם נחליף כל בדיקה כזאת בהופכית שלה, נקבל אלגוריתם הפותר את בעיה ב' באותו סדר גודל של זמן ריצה (גם בהנחה שבדיקה האם צלע קיימת או לא קיימת לוקח אותו סדר גודל זמן ריצה).
אך אלגוריתם מבוסס רדוקציה כפי שתיארנו לעיל, לא ירוץ בסדר גודל של פחות מ- $O(n^2)$, עקב המרת הקלט.

שאלה 3 סעיף א'



שאלה 3 סעיף ב' – תיאור הרדוקציה

<u>שלב תרגום הקלט:</u> עבור מופע לבעיית כפל המטריצות, כלומר מטריצות בוליאניות A ו- B , נבנה את הגרף הדו-חלקי המתאים להן כמתואר בשאלה, ונסמן אותם $G(A)$ ו- $G(B)$, ונגדיר:
$G(A)=(V\cup U, E)$ כאשר $V=v_1, v_2, .., v_n$, $U=u_1, u_2, .., u_n$
ו- $E=(v_i, u_j):M_{i,j}=1$. כנ"ל עבור $G(B)=(U\cup F, E')$, $U=u_1, u_2, .., u_n$
ו- $F=f_1, f_2, .., f_n$ וכמובן $E'=(u_i, f_j):M_{i,j}=1$. כעת, נתבונן בגרף G , המוגדר כך:
$G=(V\cup U\cup F, E\cup E')$.

<u>שלב תרגום הפלט:</u> עבור פלט המתקבל כפתרון לבעיית מטריצת מרחקים, אשר בו נקבל מטריצה מסדר $3\mathrm{n}\ X\ 3\mathrm{n}$, נחזיר מטריצה בגודל $n\ X\ n$ אשר נמצאת בפינה ימנית עליונה של המטריצה מפלט האלגוריתם – כלומר חיתוך 3 השורות הראשונות עם 3 העמודות האחרונות של המטריצה, כמו כן, נהפוך כל -1 ל- 0 ו- 2 ל- 1 .

אלגוריתם לבעיית כפל המטריצות מבוסס רדוקציה: בהינתן מופע לבעיית כפל המטריצות כמתואר לעיל, צור מטריצה G כמתואר בשלב המרת הקלט, הפעל עליה את "הקופסא השחורה" של אלגוריתם מטריצת המרחקים והמר את הפלט כמתואר לעיל.

שאלה 3 – הוכחת נכונות

<u>רעיון ההוכחה:</u> נראה כי עבור איבר $C_{i,j}$ במטריצה C , ערכו יהיה 0 אם אין מסלול בין v_i ל- f_j
בגרף G אשר התקבל מתהליך המרת הקלט, וערכו יהיה 1 אם ורק אם קיים מסלול בין v_i לבין f_j בגרף זה.
כלומר, במטריצת המרחקים עבור גרף G יופיע הערך -1 עבור (v_i, f_j) אם אין מסלול בין קודקודים אלה,

ו- 2 אם כן.

הוכחת נכונות:

טענה ראשית: במטריצת המכפלה, ערכו של איבר $C_{i,j}$ יהיה 1 אם"ם קיים מסלול בין v_i ו- f_j

וערכו יהיה 0 אם"ם לא קיים מסלול כזה.

אבחנה: אם קיים מסלול בין קודקוד v_i ל- f_j בגרף G , אורכו 2 , וכל המסלולים באותו כיוון.
הסבר: ניתן להתבונן בגרף זה כגרף "תלת-חלקי" – גרף אשר כולל 3 קבוצות קודקודים, אשר אין צלע בין שני קודקודים
מאותה קבוצה, וכמו כן יש רק צלעות מקבוצה V לקבוצה U ומקבוצה U לקבוצה F , ומכיוון שהגרף מכוון
כל המסלולים יהיו באותו הכיוון.

הוכחת הטענה הראשית: נתבונן ב- $n \geq i, j \geq 0$ כלשהם. כפל מטריצות בוליאני, המוגדר בשאלה כך:
$C_{i,j} = \vee_{k=1}^{n}\left(A_{i,k} \wedge B_{k,j}\right)$ יניב את הערך 1 אם"ם $\exists k \in N, n \geq k \geq 0$ כך ש- $A_{i,k} \wedge B_{k,j} = 1$
(לפי הגדרת OR). כלומר, אם קיימת צלע $e = \left(v_i, u_k\right)$ וגם $e' = \left(u_k, f_j\right)$ בגרף G.
אם לא קיימות זוג צלעות כזה, הרי כל ערכי הביטויים $\left(A_{i,k} \wedge B_{k,j}\right)$ יקבלו את הערך 0 ולכן ערך ה- $C_{i,j}$ יהיה
0 . נשיב לב כי לפי האבחנה לא יתכן מסלול באורך יותר מ-2, וכל המסלולים האלה באותו כיוון, ולכן אם קיים מסלול כלשהו
בין קודקוד מ-V לקודקוד מ-F, אורכו יהיה בדיוק 2. כעת, לאחר שנפעיל את אלגוריתם עבור בעיית מטריצת המרחקים, נוכל
להמיר מרחק חיובי ל-1 ומרחק שלילי(-1 , כלומר שלא קיים מסלול) ל-0, ולקבל בדיוק את מטריצת הכפל הנדרשת לפתרון.

שאלה 3 – ניתוח זמן ריצה

המרת קלט: יצירת מטריצת סמיכויות המייצגת את גרף G כמתואר לעיל ($O\left(n^2\right)$) . מעבר על המטריצות A ו-B והעתקת

הערכים המתאימים למטריצה שתייצג את G ($O\left(n^2\right)$). בסך הכל - $O\left(n^2\right)$.
הרצת האלגוריתם לפתרון בעיית מטריצת המרחקים: עבור כל קודקוד ב-G, נריץ BFS למציאת מרחקו משאר הקודקודים. BFS -

$O\left(|V'|+|E'|\right)$, ולכן $O\left(|V'| * \left(|V'|+|E'|\right)\right)$, $E' = O\left(|V'|^2\right)$ כמובן.
המרת הפלט: העתקת תת-המטריצה בגודל $n X n$ הרלוונטית והמרת הערכים - $O\left(n^2\right)$.

סך הכל זמן ריצת האלגוריתם מבוסס הרדוקציה: $O\left(|V'| * \left(|V'|+|E'|\right)\right)$.

שאלה 4 – תיאור האלגוריתם

1. $G \leftarrow ((c_1, k_1), (c_2, k_2), ..., (c_n, k_n))$
2. $S \leftarrow (0,0,..,0)$
3. $weight \leftarrow 0$
4. $while(weight < 10 \wedge G \neq \varnothing)$
   - 4.1 $find\ i \in G\ such\ that\ c_i\ is\ maximum$
   - 4.2 $diff \leftarrow min(k_i, 10 - weight)$
   - 4.3 $weight \leftarrow weight + diff$
   - 4.4 $G \leftarrow G \setminus \{i\}$
   - 4.5 $S[i] \leftarrow diff$
5. $return\ S$

בכל איטרציה של הלולאה נבחר את התבלין היקר ביותר מהתבלינים שנותרו בחנות ומוסיפים לשק את הכמות האפשרית מתבלין זה (עד 10 ק"ג).

שאלה 4 – הוכחת נכונות

<u>אינטואיציה:</u> עבור הגנב, אין הבדל בין התבלינים מעבר לערכם, ולכן ירצה למלא את השק בתבלין בעל הערך הגבוה ביותר בחנות, כאשר זה יגמר יעבור אל התבלין הבא הכי יקר, וכך הלאה עד אשר ימלא את השק או יגמרו התבלינים בחנות.

<u>טענה ראשית:</u> הגנב ימלא את השק כך שערכם של התבלינים בשק יהיה המקסימלי.
<u>טענת עזר:</u> בכל שלב של האיטרציה, קיים פתרון חוקי ואופטימלי O אשר מכיל את כל התבלינים שבשק ובאותן כמויות.

אבחנה א': בכל פעם שהגנב מוסיף תבלין לשק, תבלין זה בעל ערך מקסימלי מבין התבלינים שנותרו בחנות.
הסבר: האלגוריתם בוחר את התבלין הבא לפי ערכו בחנות.
אבחנה ב': כל פתרון בו האלגוריתם יבחר הוא פתרון חוקי.
הסבר: בכל פעם שהגנב מוסיף תבלין לשק, הוא מוסיף לכל היותר k_i ק"ג לשק, אך לא יותר מהכמות הפנויה בשק.

<u>הוכחת הטענה הראשית:</u> בסוף האלגוריתם, נותרנו עם שק התבלינים שהגנב בחר. אם לא נותרו תבלינים בחנות, הרי פתרון זה הפתרון היחיד ולכן אופטימלי מאבחנה ב' גם חוקי ולכן סיימנו.
אם נותרו תבלינים בחנות, לפי טענת העזר הרי התבלינים בשק והכמויות שבחרנו מוכלים בפתרון חוקי ואופטימלי כלשהו, ולכן פתרון זה יהיה חוקי, אופטימלי וערכו כמובן יהיה מקסימלי, כנדרש.

<u>הוכחת טענת העזר:</u> באינדוקציה על כמות התבלינים בשק.
<u>הגדרה:</u> נאמר **שפתרון S מוכל בפתרון O** אם לכל $n \geq i \geq 1$ מתקיים: $S[i]=0 \vee S[i]=O[i]$
<u>בסיס האינדוקציה:</u> $m=0$ ולכן S מוכל בכל פתרון ובפרט בפתרון אופטימלי O (המובטח לנו מהערה

בתרגיל).
<u>הנחת האינדוקציה:</u> S_{m-1} (הסדרה S שהתקבלה לאחר $m-1$ איטרציות) מוכלת בפתרון אופטימלי O כלשהו.
<u>צעד האינדוקציה:</u> נסמן: i התבלין שנבחר באיטרציה ה-m ו- d_i הכמות שנבחרה מתבלין זה.
מקרה א' – $d_i = O_i$, ולכן לפי הנחת האינדוקציה בהכרח S_m מוכל בפתרון אופטימלי O כלשהו.
מקרה ב' – $d_i \neq O_i$, נעזר בעקרון ההחלפה – נתבונן בכמות O_i - הכמות מהתבלין ה- i בפתרון האופטימלי. כמות זו בהכרח קטנה מהכמות d_i שנבחרה ע"י האלגוריתם, כיוון שכך בנוי האלגוריתם – הוא ייקח את הכמות המקסימלית שניתן לבחור מהתבלין אותו בחר להוסיף לשק.
בנוסף, ב- $O \setminus S_{m-1}$ בהכרח קיים תבלין בעל עלות זהה לתבלין ה- i - אחרת, שב-O נמצא תבלין שערכו נמוך יותר מערכו של התבלין ה- i בעוד שנשאר לקחת עוד מהתבלין ה- i , בסתירה למקסימליות O.
לכן, מקיום תבלין (j) זה (בעל עלות זהה לתבלין שנבחר) נקזז ממנו את $d_i - O_i$. במידה ו O_j קטן מהפרש זה, בצורה דומה קיים תבלין נוסף בעל אותו ערך ממנו נוכל לקזז את יתר ההפרש.
כלומר, יצרנו פתרון חוקי ומקסימלי O' כך ש- S_m מוכלת בו.

שאלה 4 – ניתוח זמן ריצה

מיון התבלינים בסדר יורד לפי ערכם - $O(nlogn)$ עבור מציאת התבלין בעל ערך מקסימלי ב $O(1)$.

האלגוריתם מבצע לכל היותר n איטרציות, כל איטרציה מתבצעת ב- $O(1)$ זמן,

ולכן סך הכל האלגוריתם רץ בזמן של $O(n) + O(nlogn) = O(nlogn)$.