# אלגוריתמים – פתרון תרגיל 3

1. נתון גרף לא מכוון $G=(V,E)$ ותת קבוצה $W\subseteq V$, ונתונים שני צמתים $s,t\in V$. תארו אלגוריתם יעיל למציאת מסלול מ- s ל- t המבקר במספר מינימלי של צמתים מ- W.

**תיאור האלגוריתם:**

א. נגדיר פונקצית משקל $w:E\to R$ כך:

$$w((v,u))=\begin{cases}2 & v\in W\wedge u\in W\\ 1 & v\in W\vee u\in W\\ 0 & otherwise\end{cases}$$

ב. נריץ את הגרסה הליניארית שראינו בתרגול של האלגוריתם של דייקסטרה למציאת מסלול קצר ביותר החל מצומת s (ניתן לעשות זאת כיוון שמשקל הקשתות הוא 2- 0).

ג. בתום ריצת האלגוריתם של דייקסטרה, נפנה לצומת t. אם d[t] שווה לאינסוף נחזיר "אין" אחרת נשתמש במצביעי $\pi$ כדי להחזיר את המסלול בין s ל-t. זהו המסלול המכיל מספר מינימאלי של צמתים מ-W.

**הוכחת נכונות:**

נשים לב שאם מוחזר "אין" אזי לפי הנכונות של אלגוריתם דייקסטרה, לא קיים מסלול בין s ל-t.

נראה כי המסלול המוחזר הוא המסלול העובר דרך מספר צמתים מינימאלי מ-W. נניח בשלילה שקיים מסלול המכיל פחות צמתים מ-W. לכן, בפרט, קיים צומת $v_k\in W$ כך שניתן "לעקוף" אותו דרך מסלול $v_{k-1}, u_1,...u_m, v_{k+1}$ כך ש-$u_1,...u_m$ אינם ב-W (נזכור שהגרף אינו מכוון לכן במקרה הצורך נוכל אף "לחזור" על עקבותינו).

משקל המסלול $u_1,...u_m$ הוא 0 (כיוון שכל הצמתים אינם ב-W). לכן, משקל המסלול $v_{k-1}, u_1,...u_m, v_{k+1}$ הוא בין 0 ל-2 (בהתאם לשייכות של $v_{k-1}$ ו-$v_{k+1}$ ל-W) ובכל מקרה, קטן ממש משקל המסלול המקורי. אך המסלול המוחזר מאלגוריתם דייקסטרה הוא בהכרח המסלול הקצר ביותר. קיבלנו סתירה ולכן המסלול המוחזר עומד בדרישות.

**ניתוח סבוכיות:**

א. הרצת הגרסה הליניארית של האלגוריתם של דייקסטרה – $O(|V|+|E|)$

ב. החזרת המסלול בין s ל-t (במידה וקיים) - $O(|V|+|E|)$

סה"כ סבוכיות זמן הריצה של האלגוריתם – $\mathbf{O(|V|+|E|)}$.

**מ.ש.ל**

2. נתון גרף מכוון $G=(V,E)$ עם פונקצית משקל $w:E\to R$, וקודקוד $s\in V$. נגדיר אורך של מסלול להיות משקל הקשת הכבדה ביותר במסלול. תארו אלגוריתם יעיל למציאת מסלולים קצרים ביותר מ- s לכל $v\in V$ לפי הגדרה זו.

**תיאור האלגוריתם:**

נגדיר פונקצית Relax חדשה:

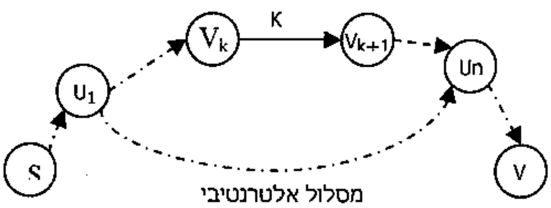
```
Relax(u,v) = if d[v] > max(d[u], w(u,v)) then
                 d[v] <- max(d[u], w(u,v))
                 π[v] <- u
```

כלומר, בעת ביצוע relax אנו בוחרים את המינימום בין ערך d המקורי של צומת היעד לבין ערך המקסימום בין משקל הקשת (u,v) לערך d של צומת המקור.
כמו כן, נאתחל את ערכי ה-d של כל הצמתים המחוברים ישירות ל-s להיות משקל הקשת בינם לבין s ואת מצביעי ה- $\pi$ להצביע על s. את s נאתחל כרגיל.
נשים לב שכיוון שאין פה פעולת חיבור, אין משמעות למעגלים שליליים כאן (כי בדיקתם היא סופית ולא תמשך בלולאה אינסופית). לכן, נוכל ונריץ אלגוריתם דייקסטרה למציאת המסלולים הקצרים ביותר החל מצומת s על הגרף עם הגדרת האתחול החדשה והגדרת

relax החדשה. הערכים שיתקבלו עבור כל צומת בתם האלגוריתם ייצגו את המסלול הקצר ביותר ל-s על פי ההגדרה החדשה.

**הוכחת נכונות:**

נניח בשלילה שקיים צומת v כך ש- $d[v] > \delta(s,v)$.




נסמן את המסלול שהחזיר האלגוריתם ב- $s,v_1,v_2,...,v$. כמו כן, נסמן את משקל הקשת ה"כבדה" ביותר במסלול מ-s ל-v כ-K. נניח בה"כ שקשת $(V_k,V_{k+1})$ היא הקשת היחידה במסלול שהחזיר האלגוריתם מ-s ל-v במשקל זה (שאר הקשתות במשקל נמוך ממש מ-K). ניתן להניח זאת כי הסבר עבור קשת בודדת תקף עבור כל קשת אחרת במשקל המקסימאלי.

ע"פ ההנחה בשלילה, קיים מסלול אחר מ-s ל-v שאינו מכיל את הקשת $(V_k,V_{k+1})$ נניח שהמסלול הוא $s,v_1,...,u_1,...u_n,...v$ כאשר $u_n$ הוא הצומת הראשון המשותף לשני המסלולים לאחר הפיצול. נשים לב שמצב זה אינו אפשרי כיוון כאשר האלגוריתם יגיע לקשתות המתחברות אל $u_n$, אם הקשת המגיעה מהמסלול עם $(V_k,V_{k+1})$ נבדקה ראשונה, הרי שיהיה שיפור ולכן האלגוריתם יעדכן את $u_n$ להצביע על המסלול האלטרנטיבי כטוב יותר. אם המסלול האלטרנטיבי נבדק קודם, לא יהיה שיפור ולכן האלגוריתם ישאיר את הצומת $u_n$ להצביע על המסלול האלטרנטיבי. קיבלנו סתירה ולכן המסלול המוחזר הוא אכן הקצר ביותר ע"פ ההגדרה החדשה.

**ניתוח סבוכיות:**

פעולת האתחול החדשה מתבצעת ב-O(|V|). פעולת ה-Relax החדשה מתבצעת בזמן O(1). פרט לכך אין שינויים באלגוריתם דייקסטרה ולכן סבוכיות הפתרון היא כסבוכיות אלגוריתם דייסקטרה. כלומר, **O(|V|log|V| + |E|)**.

**▪ מ.ש.ל**

3. נתון גרף מכוון G=(V,E) עם פונקצית משקל חיובית $w:E\rightarrow R^+$, וקודקוד $s\in V$, ונתונה פונקציה $f:V\rightarrow R^+$. תארו אלגוריתם לינארי שבודק האם לכל $v\in V$ מתקיים $f(v) = \delta(s,v)$.

**תיאור האלגוריתם:**

א. תחילה נבדוק האם f(s)=0. אם לא, נחזיר F.

ב. נריץ BFS החל מצומת s. לאחר ריצת BFS נבדוק האם כל הצמתים נגישים מ-s (נבדוק שלכולם ערך d הקטן מאינסוף). אם לא, נעצור את האלגוריתם ונחזיר F (כיוון שf מוגדרת על הממשיים החיוביים ולכן חייבת להחזיר מספר כלשהו (אינסוף אינו מוגדר כ"מספר" ב-R+)).

ג. נעבור על כל הקשתות בגרף. לכל קשת (u,v) נבדוק האם f(u)+w(u,v) > f(v). אם כן, נמחק את הקשת מן הגרף. כמו כן, נבדוק האם f(u)+w(u,v) < f(v). אם כן, נעצור את האלגוריתם ונחזיר F.

ד. בשלב זה נמצאים בגרף רק קשתות שעבורן מתקיים f(u)+w(u,v)=f(v). נריץ שוב BFS החל מצומת s. לאחר ריצת BFS נבדוק האם כל הצמתים נגישים מ-s (נבדוק שלכולם ערך d הקטן מאינסוף). אם לא, נעצור את האלגוריתם ונחזיר F. אם כל הצמתים נגישים מ-s נחזיר T.

**הוכחת נכונות:**

כיוון שפונקציית המשקל מכילה משקלים חיוביים בלבד, המרחק המינימאלי אל s (מ-s) הוא כמובן 0. לכן, אם f(s) אינו 0, ברור שהתכונה אינה מתקיימת ואנו מחזירים F כנדרש.
כמו כן, כיוון שf מחזירה ערכים מהממשיים החיוביים (ואינסוף אינו מספר ממשי חיובי), אם קיימים צמתים שאינם נגישים מ-s הרי שדלתא שלהם תהיה אינסוף ולכן f בוודאי לא תתאים לדרישה. גם במקרה זה אנו מחזירים F כנדרש.

לאחר הבדיקות שביצענו בסעיף ג', עבור כל הקשתות שנשארו בגרף מתקיים f(v)=f(u)+w(u,v). לכן, כל מסלול שנשאר בגרף הוא במשקל f(v), כלומר מתקיים $f(v) \geq \delta(s,v)$. נוכיח באינדוקציה כי מתקיים גם $f(v) \leq \delta(s,v)$.

<u>שלב א</u> – שלב הבדיקה: עבור v=s מתקיים $f(s) = \delta(s,s) = 0$.

<u>שלב ב</u> – שלב המעבר
יהי v (שאינו s) צומת בגרף. ויהי u הצומת שלפני v. ע"פ הנחת האינדוקציה מתקיים $\delta(s,u) \geq f(u)$. כמו כן, בגרף המסלולים הקצרים מתקיים $\delta(s,v) = \delta(s,u) + w(u,v)$.
סה"כ קיבלנו כי $\delta(s,v) = \delta(s,u) + w(u,v) \geq f(u) + w(u,v) = f(v)$.

הראנו כי מתקיים $f(v) \geq \delta(s,v)$ וגם מתקיים $f(v) \leq \delta(s,v)$ ולכן מתקיים $f(v) = \delta(s,v)$.

**ניתוח סבוכיות:**

א. בדיקת ערך f עבור s – O(1)

ב. הרצת BFS פעמיים מצומת s – O(|V|+|E|)

ג. בדיקה פעמיים של ערכי d של הקשתות – O(|E|)

ד. בדיקת ערכי הקשתות בגרף, ביצוע ההשוואה ותגובה בהתאם – O(|E|)

סה"כ סבוכיות זמן ריצת האלגוריתם – **O(|V|+|E|)**

**▪ מ.ש.ל**

4. נתון גרף לא מכוון G=(V,E) עם משקלים אי שליליים על הקשתות, כל קשת צבועה באדום או בכחול, ונתון קודקוד s∈V. תארו אלגוריתם יעיל המוצא, לכל v∈V, מסלול קצר ביותר מ-s ל- v מבין המסלולים שמכילים 3 קשתות אדומות לכל היותר (ומספר כלשהו של קשתות כחולות).

**תיאור האלגוריתם:**

א. תחילה נחליף את כל קשת (u,v) בזוג קשתות מכוונות (u,v) ו-(v,u). משקל הקשתות החדשות יהיה כמשקל הקשת המקורית.

ב. נבצע שכפול של קודקודי הגרף ל-4 קבוצות $V_0=V_1=V_2=V_3=V$.

ג. נעבור על כל קשת (u,v) בגרף המקורי. אם הקשת צבועה בכחול, ניצור 4 העתקים שלה: $(u_0,v_0), (u_1, v_1), (u_2, v_2), (u_3, v_3)$. כלומר, כל קשת כחולה תחבר את העתקים של 2 הקודקודים בכל קבוצה. אם הקשת בצבע אדום, ניצור 3 העתקים כך שכל עותק יחבר בין 2 קבוצות: $(u_0,v_1), (u_1, v_2), (u_2, v_3)$. נשים לב שחיבור הקשתות האדומות לא יאפשרו "חזרה" לקבוצה קודמת.

ד. נריץ אלגוריתם דייקסטרה בגרף החדש שבנינו החל מהייצוג של הצומת s בקבוצה $V_0$ (כלומר החל מצומת $s_0$).

ה. לאחר סיום ריצת האלגוריתם, אורך המסלול הקצר ביותר לכל אחת מהצמתים הוא המינימום מבין אורך המסלול לכל צומת בכל אחת מ-4 הקבוצות.

**הוכחת נכונות:**
תחילה נשים לב שכיוון הקשתות אינו פוגע בקשירות הגרף כיוון שבמקום כל קשת לא מכוונת אנו מוסיפים 2 קשתות מכוונות בכיוונים מנוגדים. כמו כן, עלות הקשתות נשמרת כיוון שהקשתות החדשות "שוקלות" כמו הקשת המקורית וכן מעבר בקשת נעשה בכל פעם לכיוון יחיד (כך שאם עברנו ב2 הקשתות לשני הכיוונים זה שקול לכך שעברנו פעמיים בקשת המקורית ולכן המשקל נשמר).

אנו מריצים את אלגוריתם דייסקטרה על הגרף החדש, לכן בגרף החדש המרחק של כל צומת מ$s_0$ השמור ב-d של הצומת הוא המרחק המינימאלי. מההגדרות נובע שלכל צומת v קיים מסלול מ-s עם פחות מ-3 קשתות אדומות אמ"מ בגרף החדש קיים מסלול מ-s לאחד העתקים של v. הבחירה של המסלול המינימאלי מבין 4 המסלולים האפשריים מבטיחה לנו שמסלול זה יהיה גם המינימאלי בגרף המקורי (המכיל לכל היותר 3 קשתות אדומות). נשים לב שהסיבה לכך היא שההבדל בין הקבוצות השונות הוא במספר הקשתות האדומות שהשתמשנו עד כה. חלוקה זו מאפשרת מציאת מסלול קצת ביותר תוך הגבלה על מספר הקשתות האדומות.

**ניתוח סבוכיות:**

א. החלפת הקשתות בזוג קשתות מכוונות – $O(|E|)$.

ב. שכפול קודקודי הקשתות ל-4 העתקים – $O(|V|)$.

ג. הגדרת הקשתות בגרף החדש ע"פ הצבע - $O(|E|)$=$O(4|E|)$

ד. ריצת דייקסטרה על הגרף החדש – $O(|V|\log|V| + |E|)$=$O(4|V|\log4|V| + 4|E|)$

ה. מציאת המסלול הקצר ביותר עבור כל צומת (מתוך 4) – $O(|V|)$

סה"כ סיבוכיות זמן ריצת האלגוריתם – **$O(|V|\log|V| + |E|)$**

**▪ מ.ש.ל**

5. נתון גרף מכוון $G=(V,E)$ עם פונקצית משקל $w:E\rightarrow R$, וקודקוד $s\in V$. ידוע שאין בגרף מעגלים מכוונים שליליים, ושלכל $v\in V$ קיים מסלול קצר ביותר מ- s ל- v המשתמש ב- m קשתות לכל היותר. תארו אלגוריתם שרץ בזמן $O(m|E|)$ ומוצא מסלולים קצרים ביותר מ- s לכל $v\in V$.

**תיאור האלגוריתם:**
נשתמש בווריאציה של האלגוריתם של בלמן-פורד. נאתחל את הצמתים כרגיל. לאחר מכן, במקום לרוץ על כל הקשתות $|V|-1$ פעמים, נרוץ m פעמים בלבד. כמו כן, כיוון שנתון שבגרף אין מעגלים שליליים, לא נבדוק בסוף האלגוריתם האם קיים בגרף מעגל שלילי. לבסוף נחזיר True.

**הוכחת נכונות:**
נתון כי כל מק"ב בגרף הוא לכל היותר באורך m. בכל פעם שאנו מבצעים את נסיונות השיפור בלולאה על כל הקשתות, אנו משפרים לאורך הקצר ביותר, במקרה הגרוע ביותר לפחות צומת אחד במסלול כל מק"ב. כיוון שנתון שאורך כל מסלול הוא לכל היותר m קשתות, לאחר m לולאות שיפור, הגענו בוודאות למצב הטוב ביותר האפשרי. נשים לב, שבהוכחה המקורית לנכונות בלמן פורד, אנו רצים $|V|-1$ פעמים כיוון שגודל המסלול המקסימאלי הוא לכל היותר $|V|-1$. לכן, בהסתמך על ההוכחה המקורית, במקרה ואנו יודעים שלכל היותר אורך כל מסלול הוא m, מספיק אכן לבצע m הרצות של לולאת השיפור.

**ניתוח סבוכיות:**
א. אתחול s ושאר הצמתים – $O(|V|)$.
ב. הרצת לולאת השיפור m פעמים – $O(|mE|)$.

כיוון שלכל צומת יש קשר לצומת s, קיימות לפחות $|V|$ קשתות (לכל צומת חייבת להיות לפחות קשת אחת מחוברת אליו). לכן, סה"כ סבוכיות זמן הריצה – **$O(m|E|)$**.

**▪ מ.ש.ל**

# אלגוריתמים – תרגיל 3

<u>הנחיה כללית</u>: עבור כל אחת מהשאלות שבהן נדרש למצוא אלגוריתם יש להוכיח את נכונותו ולנתח את סיבוכיות זמן הריצה שלו.

1. נתון גרף לא מכוון $G=(V,E)$ ותת קבוצה $W\subseteq V$, ונתונים שני צמתים $s,t\in V$. תארו אלגוריתם יעיל למציאת מסלול מ- s ל- t המבקר במספר מינימלי של צמתים מ- W.

2. נתון גרף מכוון $G=(V,E)$ עם פונקצית משקל $w:E\rightarrow R$, וקודקוד $s\in V$. נגדיר אורך של מסלול להיות משקל הקשת הכבדה ביותר במסלול. תארו אלגוריתם יעיל למציאת מסלולים קצרים ביותר מ- s לכל $v\in V$ לפי הגדרה זו.

3. נתון גרף מכוון $G=(V,E)$ עם פונקצית משקל חיובית $w:E\rightarrow R^+$, וקודקוד $s\in V$, ונתונה פונקציה $f:V\rightarrow R^+$. תארו אלגוריתם לינארי שבודק האם לכל $v\in V$ מתקיים $f(v) = \delta(s,v)$.

4. נתון גרף לא מכוון $G=(V,E)$ עם משקלים אי שליליים על הקשתות, כל קשת צבועה באדום או בכחול, ונתון קודקוד $s\in V$. תארו אלגוריתם יעיל המוצא, לכל $v\in V$, מסלול קצר ביותר מ- s ל- v מבין המסלולים שמכילים 3 קשתות אדומות לכל היותר (ומספר כלשהו של קשתות כחולות).

5. נתון גרף מכוון $G=(V,E)$ עם פונקצית משקל $w:E\rightarrow R$, וקודקוד $s\in V$. ידוע שאין בגרף מעגלים מכוונים שליליים, ושלכל $v\in V$ קיים מסלול קצר ביותר מ- s ל- v המשתמש ב- m קשתות לכל היותר. תארו אלגוריתם שרץ בזמן $O(m|E|)$ ומוצא מסלולים קצרים ביותר מ- s לכל $v\in V$.

6. נתונה מערכת אי-שיוויונים במשתנים $x_1, \ldots, x_n$. כל אי-שיוויון הוא מהצורה $x_i \cdot x_j \le c_{ij}$. כתבו אלגוריתם שמוצא פתרון למערכת או מחזיר ש"לא קיים כזה".

BF

7. <u>שאלת בונוס (10 נקודות):</u>
נתון גרף מכוון $G=(V,E)$ עם פונקצית משקל $w:E\rightarrow\{0,1,2,\ldots k\}$, ונתון קודקוד $s\in V$. יש למצוא מק"בים מ-s לכל אחד מקודקודי V בזמן $O(\,k(|V|+|E|)\,)$, תוך שימוש ב- $O(|V|+k)$ זיכרון נוסף (ולא ב- $O(k|V|)$ זיכרון נוסף כפי שראינו בתירגול).

$m \le |V|-1$

$c_{ij} \ge x_i x_j$

max(

$x_i x_j \le c_{ij}$