תרגול 1- מבוא ורדוקציות

מבוא והגדרות בסיסיות

דיון במושג הבעיה:

נתחיל בדוגמה לבעיה: ברשותנו מפת כבישים, ועלינו להגיע מעיר א' לעיר ב', וברצוננו למצוא מסלול קצר ביותר בין שתי הערים.

בבעיה נשאלת שאלה מסוימת קבועה, אשר נשאלת על קלט משתנה הנקרא **מופע** (instance). במקרה הנ"ל, המופע (או הקלט) כולל את מפת הכבישים, עיר המוצא ועיר היעד הנמצאות על המפה. **פתרון** הוא תשובה לשאלה המוצגת בבעיה עבור מופע מסוים. הפתרון במקרה הנ"ל יהיה תיאור של מסלול המתחיל בעיר א', עובר דרך כבישים המסומנים במפה הנתונה, ומסתיים בעיר ב' (והוא קצר ביותר מבין כל המסלולים מסוג זה).
לעתים נותנים לבעיה **שם**, למשל במקרה זה "בעיית המסלול הקצר ביותר".

הגדרת המושג אלגוריתם:

דוגמה - אלגוריתם ל"בעיית המסלול הקצר": בהינתן מפה ושתי ערים, בדוק כל מסלול אפשרי במפה בין שתי הערים אשר אינו מבקר באף עיר יותר מפעם אחת, והחזר מסלול קצר ביותר מבין כל המסלולים הנ"ל, או החזר "אין פתרון" במידה ולא קיים מסלול בין שתי הערים.

אלגוריתם עבור בעיה מסוימת הוא סכמה המתארת סידרה של פעולות אשר מתבצעות על מופע של הבעיה, ובסיומו מוחזר פלט עבור המופע הנתון. פלט זה יכול להיות פתרון למופע, או הכרזה כי "לא קיים פתרון למופע זה של הבעיה". אלגוריתם עבור בעיה חייב להחזיר פלט נכון לכל מופע שיקבל כקלט (כלומר להחזיר פתרון אם קיים, או להחזיר את הפלט "לא קיים פתרון" אם לא קיים פיתרון).

הוכחת נכונות פורמאלית:

כיצד נדע האם האלגוריתם שהצגנו אכן פותר את הבעיה המבוקשת? עלינו להוכיח זאת! **הוכחת נכונות פורמלית** של טענה מסוימת היא סדרה של טענות, אשר כל אחת מהן נגררת לוגית אך ורק מתוך ידע מוקדם מוכח ומתוך הנחות או טענות מוקדמות מהקלט של הבעיה, כך שהמסקנה המבוקשת היא הטענה האחרונה בסדרה.
לדוגמא: נתון אלגוריתם אשר "לכל מפת כבישים, עיר מוצא ועיר יעד, האלגוריתם יחזיר מסלול קצר ביותר במפה בין שתי הערים". עלינו להוכיח כי האלגוריתם אכן מקיים את הטענה לעיל תחת שני מקרים:

א. ישנו מסלול במפת הכבישים בין שתי ערים נתונות.
ב. במפת הכבישים אין מסלול בין שתי ערים נתונות.

<u>בעיות הכרעה ובעיות אופטימיזציה:</u>

נתבונן בבעיות הבאות:

<u>בעיה A</u>: בהינתן גרף G ושני קודקודים בגרף: u ו- v, האם קיים מסלול בין u ל-v ב-G?

<u>בעיה B</u>: בהינתן גרף G ושני קודקודים בגרף: u ו- v, מצא מסלול קצר ביותר בין u ל-v ב-G, או החזר "אין מסלול" אם לא קיים מסלול כזה.

נבחין בין שני סוגים של בעיות: בעיות הכרעה ובעיות אופטימיזציה.
בעיות הכרעה הן כאלה שלכל מופע אנו מעוניינים לדעת <u>האם קיים פתרון או לא קיים פתרון</u>, ולמעשה הן מנוסחות כשאלות כן/לא (ולפעמים מתייחסים לפתרון לבעיית הכרעה כאל 1 או 0).
בעיות אופטימיזציה הן כאלה שלכל מופע יתכנו מספר כלשהו של פתרונות (יתכן כי המספר הוא 0), ואנו <u>מעוניינים בפתרון אשר הוא בעל ערך אופטימאלי</u> (גדול ביותר, קטן ביותר, וכד') מבין כל הפתרונות האפשריים (כאשר ערך ההחזרה הוא "אין פיתרון" במידה ואין אף פתרון חוקי).

הבעיה A היא דוגמה לבעיית הכרעה (שאלת כן/לא), ואילו הבעיה B היא דוגמה לבעיית אופטימיזציה (יתכנו מספר מסלולים בין u ו-v, ואנו מחפשים מסלול שהוא קצר ביותר. נשים לב כי יתכנו מספר מסלולים קצרים ביותר, אך ניסוח זה של הבעיה מחפש אחד מהם בלבד.

<u>נזכר בהגדרת המושג</u> $O(...)$

אם $f(x)\in O(g(n))$, אזי קיימים $c>0$ ו- n_0, כך ש- $\forall n>n_0 : f(n)\le c\cdot g(n)$.

להלן כמה דוגמאות לזמני ריצה, כאשר n מסמל את גודל המופע:

1. $O(1)$ - זמן ריצה <u>קבוע</u>. בעיות שניתנות לפיתרון בזמן קבוע (כלומר אינו תלוי בגודל הקלט). לרוב, אינן מעניינות במיוחד.
2. $O(\log n)$ - זמן לוגריתמי.
3. $O(n)$ - זמן <u>ליניארי</u>.
4. $O(n^c)$, עבור $c\in N$ - זמן פולינומיאלי.
5. $O(2^{cn})$ - זמן אקספוננציאלי עבור $c\in N$ כלשהו. זמן זה נחשב ללא יעיל במיוחד, כאשר הזמן הדרוש לפתרון של קלטים קטנים יחסית יכול להיות עצום.

בקורס זה נעסוק בדרכים לתאר בעיות וטכניקות לפתרון יעיל של בעיות – כלומר אלגוריתמים. נפתח שפה משותפת אשר תתאר הן בעיות, והן את האלגוריתמים לפתרונן, באופן ברור ותמציתי, ותאפשר לנו להבין אחד את השני ללא טעויות. השימוש שנעשה בפורמליזם אולי ייראה בהתחלה קצת מכביד ולא "טבעי", אך ככל שמתרגלים אליו יותר הוא מקל על היכולת שלנו להגות אלגוריתמים נכונים ויעילים, וכן על היכולת שלנו להבין אחרים ולהסביר להם את עצמנו.

בבואנו לכתוב אלגוריתם הפותר בעיה נתייחס לשלושה אספקטים:

- תיאור דרך כללית להגעה לפתרון עבור מופע כלשהו של הבעיה (ע"י תיאור אלגוריתם).
- הוכחת נכונות הדרך שתיארנו (על ידי הוכחה פורמאלית).
- ניתוח זמן הריצה הדרוש לקבלת הפתרון, בהתאם לדרך שתיארנו.

<u>מעתה ואילך, בכל פעם שנבקש פתרון לבעיה מסוימת, ברירת המחדל היא לפרט את שלושת האספקטים לעיל, אלא אם יצוין אחרת</u>.

תכנון אלגוריתם באמצעות רדוקציה

בהינתן בעיה אשר האלגוריתם לפתרונה אינו ידוע לנו, אנו מעוניינים לגלות או לזהות בעיה נוספת אשר האלגוריתם עבורה כן ידוע לנו, ולבדוק האם האלגוריתם של הבעיה הידועה מסוגל לשמש אותנו לצורך בניית אלגוריתם עבור הבעיה המקורית שניתנה לנו.

דוגמה 1: אלגוריתם לבעיית SEP מבוסס על רדוקציה לבעיית SP

הגדרה:

יהי $G=(V,E)$ גרף לא מכוון ויהיו $u,v\in V$ זוג קודקודים ב- G. נסמן ב- $d(u,v)$ את אורך המסלול המינימאלי בצלעות מקודקוד u לקודקוד v. בנוסף, נגדיר כי $d(u,v)=\infty$ אם לא קיים מסלול מ-u ל- v ב- G.

הגדרת בעיית SP (Single Source Shortest Paths):

יהי $G=(V,E)$ גרף לא מכוון ו- $s\in V$ קודקוד נתון ב- G. מצא $d(s,u)$ לכל $u\in V$.

הגדרת בעיית SEP (Single Source Shortest Even Paths):

יהי $G=(V,E)$ גרף לא מכוון ו- $s\in V$ קודקוד נתון ב- G. מצא את אורך המסלול (לפי צלעות) <u>**הזוגי**</u> המינימאלי מ-s לכל $u\in V$, או ∞ אם לא קיים כזה.

אנו נשתמש באלגוריתם הפותר את בעיית SP על מנת לייצר אלגוריתם הפותר את בעיית SEP (דוגמה לאלגוריתם עבור SP הוא BFS, המוכר לנו מהקורס במבני נתונים).

הרדוקציה:

יהי $G=(V,E)$ גרף לא מכוון, ו- $s\in V$ קודקוד נתון ב- G. יחדיו הם מהווים <u>מופע</u> עבור בעיית SEP.

<u>שלב תרגום הקלט:</u> נבנה את הגרף $G'=(V',E')$ גרף דו-חלקי (גרף שקודקודיו מחולקים לשתי קבוצות קודקודים זרות כך שכל צלע בו היא בין קודקודים מקבוצות שונות) כך ש:

- $V'=V^1\cup V^2$, כאשר $V^1=\{u^1:u\in V\}$, $V^2=\{v^2:v\in V\}$ (כלומר קבוצת הקודקודים ב-G' תכיל שני עותקים של קבוצת הקודקודים ב-G, כאשר כל קודקוד מ- V^1 ו- V^2 מתאים לקודקוד כלשהו ב-V),
- $E=\{(u^1,v^2),(u^2,v^1):(u,v)\in E,\ u^1,v^1\in V^1,\ u^2,v^2\in V^2\}$.

<u>שלב תרגום הפלט:</u> נחזיר כפלט את קבוצת המרחקים הבאים: $\forall u\in V$ נחזיר את המרחק $d(s^1,u^1)$ ב- G' כמרחק הזוגי המינימאלי בין s ל-u ב-G.

<u>**אלגוריתם לבעיית SEP המבוסס על הרדוקציה:**</u>

<u>בהינתן מופע של SEP, תרגם אותו למופע של SP ,</u>הפעל אלגוריתם לפתרון SP על המופע $< G', s^1 >$ כך ש-$s^1 \in V^1$ הוא הקודקוד המתאים ל-s בקבוצה V^1, והחזר לכל $u \in V$ את המרחק שהתקבל מתרגום הפלט.

הוכחת נכונות:

רעיון ההוכחה: נראה כי כל מסלול ב G' בין s^1 לבין v^1 הוא מסלול באורך זוגי שמתאים למסלול ב- G באותו אורך בין s ל- v. האם זה מספיק להוכחת נכונות? לא, צריך להראות כי כל מסלול זוגי ב- G בין s ל- v מתאים למסלול ב- G' באותו אורך בין s^1 ל-v^1. משתי הטענות נקבל כי אורך מסלול קצר ביותר ב- G' בין s^1 ל-v^1 שווה לאורך מסלול זוגי קצר ביותר ב- G בין s ל- v. .

<u>טענה ראשית</u>: לכל קודקוד $v \in V$, אורך המסלול הזוגי המינימאלי ב-G מהקדקוד $s \in V$ אל הקדקוד v שווה ל- $d(s^1, v^1)$ ב- G'.

<u>אבחנה</u>: אורך כל מסלול (אם קיים) מ-$s^1 \in V^1$ לקודקוד $v^1 \in V^1$ כלשהו הוא זוגי.

הסבר: בכל מסלול בגרף דו חלקי הקודקודים לאורך המסלול הם מ-V^1 ו-V^2 לסירוגין. לכן, במסלול המתחיל מקודקוד ב-V^1 ומסתיים בקודקוד מ-V^1 יש מספר אי זוגי של קודקודים, כלומר מספר זוגי של צלעות.

<u>טענת עזר</u>: קיים ב- G מסלול באורך $2m$ בין $s \in V$ ל- $v \in V$ אם"ם קיים ב- G' מסלול באורך $2m$ בין $s^1 \in V^1$ ל- $v^1 \in V^1$.

<u>הוכחת הטענה הראשית (על סמך טענת העזר)</u>: יהי $2m$ אורך המסלול הזוגי המינימאלי מ-s ל-v ב-G. מטענת העזר נובע כי קיים מסלול באורך $2m$ מ-s^1 ל-v^1 ב-G'. לכן אורך המסלול הזוגי המינימאלי ב- G' הינו לכל היותר $2m$.

נניח בשלילה כי קיים מסלול באורך קצר יותר מ- $2m$ מ-s^1 ל-v^1 ב-G'. לפי האבחנה מסלול זה הוא באורך זוגי, ולכן נסמן את אורכו ב- $2k$, כאשר $2k < 2m$. מטענת העזר נובע כי קיים מסלול מ-s ל-v באורך $2k$ ב-G, בסתירה לכך שאורך המסלול הזוגי המינימאלי ב-G הוא $2m$. ולכן $d(s^1, v^1) = 2m$ ב-G'.

נשים לב כי קיים מקרה נוסף: אם לא קיים מסלול באורך זוגי מ-s ל-v ב-G, הרי שהאלגוריתם יצטרך להחזיר את הערך ∞. נראה כי אכן $d(s^1, v^1) = \infty$ ב-G'. נניח בשלילה כי $d(s^1, v^1) \neq \infty$ ב-G', אזי קיים מסלול מ-s^1 ל-v^1 ב-G', ונגיע שוב לסתירה באופן דומה לנ"ל.

הערה: שימו לב כי בזאת הוכחה טענת הנכונות העיקרית. טענה זו נוסחה כשוויון בין שני מספרים שהם ערכי הפתרונות האופטימאליים של מופעים מתאימים לבעיות SP ו- SEP. מכיוון ששוויון הינו יחס סימטרי, הרי נובע כי מספר כלשהו (או אינסוף) הוא אורך המסלול הזוגי המינימאלי ב-G מקדקוד $s \in V$ אל לכל קודקוד $v \in V$ אם"ם מספר זה הוא גם $d(s^1, u^1)$ ב-G'. וזאת בהתאם למה שצריך להוכיח על פי הגדרת המושג רדוקציה.

<u>הוכחת טענת העזר:</u>

כיוון ראשון ($\Leftarrow$):יהי $P = (s, u_1, u_2, u_3, \dots, u_{2m-1}, v)$ מסלול באורך $2m$ בין $s \in V$ ל- $v \in V$ ב-G. נתבונן בסדרת הקודקודים $P' = (s^1, u_1^2, u_2^1, u_3^2, \dots, u_{2m-1}^2, v^1)$ ב-G', כאשר u_i^j הוא הקודקוד המתאים ל- u_i בקבוצה V^j ב-G' (כאשר $j \in \{1,2\}$). בין כל שני קודקודים סמוכים ב-P ישנה צלע ב-G, ולכן, לפי בניית הגרף, בין כל שני קודקודים סמוכים ב-P' ישנה צלע ב-G', כלומר P' הינו מסלול חוקי באורך $2m$ בין s^1 ל- v^1 ב-G'.

כיוון שני ($\Rightarrow$): יהי $P' = (s^1, u_1^2, u_2^1, u_3^2, \dots, u_{2m-1}^2, v^1)$ מסלול באורך $2m$ בין s^1 ל- v^1 ב-G'. נתבונן בסדרת הקודקודים$P = (s, u_1, u_2, u_3, \dots, u_{2m-1}, v)$ ב-G, כאשר u_i הוא הקודקוד המתאים ל- u_i^j ב-G. בין כל שני קודקודים סמוכים ב-P' ישנה צלע ב-G', ולכן לפי בניית G', בין כל שני קודקודים סמוכים ב-P ישנה צלע ב-G, כלומר P הינו מסלול חוקי באורך $2m$ בין s ל-v ב-G. <u>מ.ש.ל</u>

<u>הוכחת טענת העזר חילופית:</u> למעשה, במקרה של הוכחה זו, שני הכיוונים סימטריים, ואת הכיוון השני מספיק להסביר באופן כללי מבלי להיכנס לפרטים. כמו כן, ניסוח אפשרי אחר להוכחה אשר מראה את שני הכיוונים ביחד הוא:

סדרת הקודקודים $P = (s, u_1, u_2, u_3, \dots, u_{2m-1}, v)$ היא מסלול באורך $2m$ בין $s \in V$ ל- $v \in V$ ב-G

$\Updownarrow$ (מהגדרת המושג "מסלול")

בין כל שני קודקודים סמוכים ב-$P = (s, u_1, u_2, u_3, \dots, u_{2m-1}, v)$ ישנה צלע ב-G

$\Updownarrow$ (לפי בניית G' מתוך G, כאשר הקודקודים u_i ב-G ו- u_i^j ב-G' הם קודקודים מתאימים)

בין כל שני קודקודים סמוכים ב-$P' = (s^1, u_1^2, u_2^1, u_3^2, \dots, u_{2m-1}^2, v^1)$ ישנה צלע ב-G'

$\Updownarrow$ (מהגדרת המושג "מסלול")

סדרת הקודקודים $P' = (s^1, u_1^2, u_2^1, u_3^2, \dots, u_{2m-1}^2, v^1)$ היא מסלול באורך $2m$ בין s^1 ל- v^1 ב-G'.

(שימו לב, לא תמיד הוכחת שני הכיוונים סימטרית!)

ניתוח זמן הריצה:

הזמן הדרוש עבור תרגום הקלט הינו הזמן הנדרש לבניית הגרף G'. בכדי לבנות את G' יש לבצע מספר קבוע של מעברים על הקשתות והקודקודים של G ולכן בניית G' תדרוש $O(|V|+|E|)$.

כיון ש- $|V'| = 2|V|$, $|E'| = 2|E|$, הזמן הדרוש להפעלת BFS על G' הינו $O(|V'|+|E'|) = O(|V|+|E|)$ (ידוע מהקורס במבני נתונים).

הזמן הדרוש לתרגום הפלט דורש סריקה של קודקודי V- כלומר $O(|V|)$.

סה"כ זמן ריצת הרדוקציה כולה – $O(|V|+|E|)$.

נגדיר עתה באופן פורמאלי את המושג **רדוקציה**:

הגדרה פורמאלית ל- "רדוקציה":

תהיינה A ו-B זוג בעיות נתונות. רדוקציה מ- A לבעיה B היא זוג פונקציות f,g, כך ש:

- f היא פונקצית המרת הקלט, המעבירה מופע של בעיה A למופע של בעיה B.
- g היא פונקצית המרת הפלט, המעבירה פתרון של בעיה B לפתרון של בעיה A.
- עבור מופע a לבעיה A, אם $B\big(f(a)\big)$ הוא פתרון עבור המופע $f(a)$ תחת בעיה B אזי $g\Big(B\big(f(a)\big)\Big)$ הוא פתרון למופע a תחת בעיה A. (הגדרת נכונות)

כדי להוכיח את נכונות הרדוקציה, יש להוכיח שהאלגוריתם הבא פותר את הבעיה A:

1. עבור מופע a לבעיה A, חשב את $f(a)$.
2. עבור המופע $f(a)$ לבעיה B, חשב את הפיתרון b.
3. חשב את $g(b)$ להיות הפתרון של A.

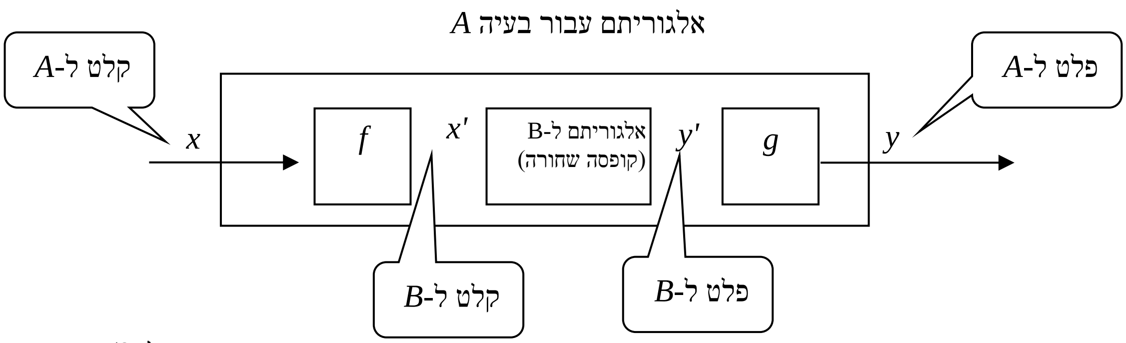
הערות לגבי ההגדרה לעיל:

א. הפונקציה f נקראת **פונקצית המרת הקלט**.

ב. הפונקציה g נקראת **פונקצית המרת הפלט**.

ג. נתייחס לאלגוריתם שפותר את בעיה B כאל "**קופסה שחורה**", מבלי להניח דבר על אופן פעולתו מלבד העובדה שהוא אכן פותר נכון את B לכל מופע.

הסכימה הבאה מסכמת את כל הנאמר לעיל:



(שאלה: בדוגמה הקודמת, מהי פונקצית הרדוקציה, מהי הקופסה השחורה, ומהי פונקצית תרגום הפלט?)

* **הערה:** לעיתים, על מנת לספק אלגוריתם הפותר את A, נרצה להשתמש מספר פעמים באלגוריתם הפותר את הבעיה B ("הקופסה השחורה"). במקרה כזה עדיין נאמר ש A ניתנת לרדוקציה ל B.

עניין אחרון עליו יש לעמוד הינו ניתוח זמן החישוב של האלגוריתם המבוסס על הרדוקציה. כפי שראינו ישנם שלושה רכיבים הדורשים זמן חישוב בעת הרצת האלגוריתם: חישוב תרגום הקלט, הפעלת האלגוריתם לפתרון הבעיה B, וחישוב תרגום הפלט. זמן חישוב האלגוריתם המבוסס על הרדוקציה הינו הזמן המקסימאלי הדרוש מבין השלושה.

הערות:

א. יש לשים לב כי בבניית הרדוקציה מבעיה A לבעיה B אנחנו לא צריכים לדעת איך פותרים את בעיה B. כלומר, אין לנו צורך בשום ידע על פעולת "הקופסה השחורה".

ב. היתרון של אלגוריתם מבוסס רדוקציה הוא שהרדוקציה יכולה להיות פשוטה יחסית (יעילה, מבחינת זמן ריצה) לאלגוריתם עבור בעיה B. אין צורך לתכנן אלגוריתם זה מחדש.

דוגמה 2: רדוקציה מבעיית SP לבעיית SEP

נראה כעת רדוקציה נוספת בכיוון ההפוך, כלומר מבעיית SP לבעיית SEP. שתי הרדוקציות שבדוגמה זו מוכיחות לנו כי שתי הבעיות שקולות.

הרדוקציה:

הגדרת מופע: יהי מופע לבעיית SP, $G=(V,E)$ גרף לא מכוון, ו- $s\in V$ קודקוד נתון ב-G.

<u>רעיון הרדוקציה:</u> נבנה G' מ G ע"י הוספת קודקוד באמצע כל צלע. כעת, כל מסלול ב G מתאים למסלול באורך כפול בגרף G' ולהפך. לכן, נוכל להריץ אלגוריתם עבור SEP ב G' ולהחזיר חצי מפלט אלגוריתם זה.

<u>תאור פורמאלי של הרדוקציה:</u>

<u>שלב תרגום הקלט:</u> נבנה את הגרף $G'=(V',E')$ גרף דו-חלקי כך:

- נבנה קבוצת קודקודים נוספת $V^*=\{\overline{uv}\mid(u,v)\in E\}$. זוהי קבוצה של קודקודי ביניים המייצגים את צלעות הגרף המקורי G. בקבוצה זו נזהה יחד קודקודים אשר נוספו עבור אותה הצלע, כלומר הקודקוד $\overline{uv}\in V^*$ והקודקוד $\overline{vu}\in V^*$ הינם למעשה אותו הקודקוד. נגדיר: $V'=V\cup V^*$.
- קבוצת הצלעות של הגרף G': $E'=\{(u,\overline{uv}):(u,v)\in E\}$. כלומר, כל קודקוד $v\in V$ יחובר בצלע לכל קודקוד מ-V^* המייצג צלע בגרף G שהייתה מחוברת לקודקוד v.

<u>פתרון הבעיה לה ביצענו רדוקציה:</u> נפעיל את אלגוריתם לפתרון SEP על המופע $<G',s>$.

<u>שלב תרגום הפלט:</u> נחזיר כפלט את קבוצת המרחקים הבאים: $\forall u\in V$ נחזיר את מחצית אורך המסלול הזוגי המינימאלי מ- $s\in V$ שחושב עבור u בגרף G', כאורך המסלול המינימאלי בין s ל- u בגרף G.

הוכחת נכונות:

צ"ל: אורך המסלול המינימאלי ב-G מקודקוד $s \in V$ אל לכל קודקוד $v \in V$ שווה למחצית אורך המסלול הזוגי המינימאלי מקודקוד s אל קודקוד v בגרף G'.

אבחנה: אורך כל מסלול ב-G' (אם קיים) מ- $s \in V$ לקודקוד $v \in V$ כלשהו הוא זוגי.

הסבר: הגרף G' שיצרנו הינו גרף דו חלקי. קודקודי V אינם מחוברים ביניהם בצלע, וכך גם קודקוד V^*. בכל מסלול אפשרי בגרף G' מופיעים קודקודים מ- V ומ- V^* לסירוגין. לכן, במסלול המתחיל מקודקוד מ-V ומסתיים בקודקוד מ-V יש מספר אי זוגי של קודקודים, כלומר מספר זוגי של צלעות.

טענת עזר: קיים ב-G מסלול פשוט באורך m בין $s \in V$ ל- $v \in V$ אם"ם קיים ב-G' מסלול פשוט באורך $2m$ בין $s \in V'$ ל-$v \in V'$.

תזכורת: מסלול פשוט הינו מסלול בו כל קודקוד מופיע לכל היותר פעם אחת. נזכיר גם כי אורך כל מסלול קצר ביותר בין שני קודקודים בגרף יהיה, למעשה, אורך המסלול הפשוט הקצר ביותר בין קודקודים אלו.

טענה: המסלול הקצר ביותר בין שני קודקודים u, v בגרף הוא מסלול פשוט.

הוכחה: יהי P המסלול הקצר ביותר בין שני קודקודים u, v בגרף ונניח בשלילה שאינו פשוט. אזי במסלול זה ישנו לפחות קודקוד אחד החוזר פעמיים (או יותר). נסמן מסלול זה $P = (u, u_2, u_3, \dots, u_i, u_{i+1}, \dots, u_j, u_{j+1}, \dots v)$. נניח כי $u_i = u_j$ הוא הקודקוד הנ"ל. אזי המסלול $P' = (u, u_2, u_3, \dots, u_i, u_{j+1}, \dots v)$ קצר יותר, בסתירה להנחה כי P הינו המסלול הקצר ביותר.

הוכחת נכונות האלגוריתם המבוסס על הרדוקציה על סמך טענת העזר:

יהי $2m$ אורך המסלול המינימאלי מ- $s \in V$ ל- $v \in V$ ב-G'. על פי טענת העזר: קיים מסלול פשוט ב- G באורך m בין $s \in V$ לבין $v \in V$. נניח בשלילה כי אורכו של מסלול זה אינו המרחק בין s ל-v ב- G. אזי קיים ב- G מסלול פשוט קצר יותר ממנו. נסמן את אורכו ב- k, כאשר $k < m$. מטענת העזר נובע כי קיים מסלול פשוט מ-s ל-v באורך $2k$ ב-G', בסתירה לכך שאורך המסלול הזוגי המינימאלי ב-G' הוא $2m$.

אי לכך, קיים מסלול מ- $s \in V$ ל- $v \in V$ ב- G באורך m ואין מסלול כזה קצר ממנו. לפיכך, $d(s,v) = m$ בגרף G.

נשים לב, שוב, כי מבנה ההוכחה יתאים גם למקרה בו לא קיים מסלול מ- $s \in V$ ל- $v \in V$ ב- G', ואז נוכיח כי $d(s,v) = \infty$ בגרף G (באופן זהה למוצג בהוכחה זו ובהוכחת הרדוקציה בכיוון ההפוך, באופן מפורט מעט יותר).

<u>הוכחת טענת העזר</u> (בקצרה):

סדרת הקודקודים $P=(s,u_1,u_2,u_3,....,u_{m-1},v)$ היא מסלול פשוט באורך m בין $s\in V$ ל- $v\in V$ ב-G.

$\Updownarrow$ (מהגדרת המושג "מסלול פשוט")

אין ב-P קודקוד החוזר פעמיים, ובין כל שני קודקודים סמוכים ב- $P=(s,u_1,u_2,u_3,....,u_{m-1},v)$ ישנה צלע ב-G.

$\Updownarrow$ (לפי בניית G' מתוך G)

אין ב-P' קודקוד החוזר פעמיים, ובין כל שני קודקודים סמוכים ב-

$P'=(s,\overline{su_1},u_1,\overline{u_1u_2},u_2,\overline{u_2u_3},....,u_{m-1},\overline{u_{m-1}v},v)$ ישנה צלע ב-G'.

$\Updownarrow$ (מהגדרת המושג "מסלול פשוט")

סדרת הקודקודים $P'=(s,\overline{su_1},u_1,\overline{u_1u_2},u_2,\overline{u_2u_3},....,u_{m-1},\overline{u_{m-1}v},v)$ היא מסלול פשוט באורך $2m$ בין $s\in V$ ל- $v\in V$ ב-G'.

על פי התזכורת, די היה להוכיח את טענת העזר על מסלולים פשוטים בכדי להוכיח את נכונות הרדוקציה למציאת מסלולים בעלי אורך מינימאלי. חישבו: מדוע, למעשה, דרשנו בטענת העזר שהמסלולים יהיו פשוטים?

רמז: כיצד תושפע ההוכחה לטענת העזר אם נאפשר במסלול P' מעבר מקודקוד $u_1\in V$ לקודקוד $\overline{u_1u_2}\in V^*$, וממנו מעבר בחזרה לקודקוד u_1 ?

ניתוח זמן הריצה:

הזמן הדרוש עבור תרגום הקלט הינו הזמן הנדרש לבניית הגרף G'. בכדי לבנות את G' יש לבצע מספר קבוע של מעברים על הקשתות והקודקודים של G ולכן בניית G' תדרוש $O\left(|V|+|E|\right)$.

נוכחנו כבר בתחילת הדוגמה, על ידי הרדוקציה מ-SEP ל-SP, כי ניתן לפתור את בעיית SEP בזמן $O(|V'|+|E'|)$ (הגרף המועבר כעת לבעיית SEP הוא הגרף G'=(V',E')). זהו החסם בו נשתמש לניתוח זמן הריצה של רדוקציה זו. כמו כן, נשים לב כי $|V'|=|V|+|E|$, וכן $|E'|=2|E|$. כלומר $O(|V'|+|E'|)=O(|V|+|E|)$. לכן, הזמן הדרוש להרצת הקופסה השחורה שתפתור את SEP הוא $O(|V|+|E|)$.

הזמן הדרוש לתרגום הפלט דורש סריקה של קודקודי V- כלומר $O\left(|V|\right)$.

סה"כ זמן ריצת האלגוריתם המבוסס על הרדוקציה הוא – $O\left(|V|+|E|\right)$.

<u>בעיות נוספות למחשבה</u>

הגדרת בעיית SOP (Single Source Shortest Odd Paths):

יהי $G=(V,E)$ גרף לא מכוון ו- $s\in V$ קודקוד נתון ב-G. מצא את אורך המסלול (לפי צלעות) **<u>האי-זוגי</u>** המינימאלי מ-s לכל $u\in V$, או ∞ אם לא קיים כזה.

1. חישבו שוב על הרדוקציה מ SEP ל SP. מה נשנה ברדוקציה על מנת להתאימה לרדוקציה מבעיית SOP לבעיית SP?

רמז: ניתן להותיר את פונקצית תרגום הקלט בדיוק כמו בדוגמה לרדוקציה שניתנה מבעיית SEP לבעיית SP.

אלו פרטים נצטרך לשנות בהוכחת הנכונות של הרדוקציה המקורית על מנת להתאימה לרדוקציה החדשה מבעיית SOP לבעיית SP ?

2. חישבו שוב על הרדוקציה מ SP ל SEP. כיצד נוכל לשנותה על מנת שתהיה זו רדוקציה מבעיית SP לבעיית SOP ?

רמז: נסו להוסיף לבניית הגרף 'G בפונקצית תרגום הקלט קודקוד נוסף, שיחובר בצלע לקודקוד s (להזכירכם: קודקוד s הינו נתון במופע לבעיית SP). כיצד נשנה את פונקצית תרגום הפלט בהתאם? אלו שינויים נערוך באבחנות ובטענות (וכמובן בהוכחתן) של הוכחת נכונות הרדוקציה?