עבודה 2 – דף תשובות

תאריך הגשה: 22/11/19, בשעה 7:59 בבוקר. יש להגיש את העבודה במערכת ההגשה.

מתרגל אחראי: נתי פטר.

הוראות כלליות:

- כל עוד לא נאמר אחרת, כאשר הנכם מתבקשים לתאר אלגוריתם יש לספק:
  1. תיאור מילולי של האלגוריתם
  2. הוכחת נכונות
  3. ניתוח זמן-ריצה
- אלגוריתם עם זמן-ריצה אקספוננציאלי נחשב לא-יעיל ולכן בדרך כלל לא יתקבל.
- יש לכתוב את הפתרון רק בדף התשובות הנלווה לעבודה.
- בכל שימוש במשפט שהוכח בכיתה, יש לצטט את המשפט באופן מדויק.

שאלה 1

סעיף א'

עבור n=30 האלגוריתם החמדן המתואר בשאלה יחזיר S={25,1,1,1,1,1} בעוד {10,10,10} הינו פתרון חוקי
בעל מספר מטבעות קטן יותר מזה שהאלגוריתם החזיר.
מצאנו מופע עבורו האלגוריתם אינו מחזיר פתרון אופטימלי.

סעיף ב'

<u>טענה ראשית:</u> האלגוריתם החמדן המתואר בשאלה מחזיר פתרון אופטימלי לבעיה.

<u>טענה נשמרת:</u> אם S_i הינה המולטי קבוצה המתקבלת לאחר הבחירה ה – i של האלגוריתם, אזי קיים פתרון אופטימלי O לבעיה כך ש - $S_i \subseteq O$.

<u>אבחנה:</u> לכל מספר טבעי n קיים פתרון חוקי (n מטבעות של 1 סנט) ולכן קיים פתרון אופטימלי.

<u>טענת עזר:</u> לכל פתרון אופטימלי כלשהו O, אם x,y,z הינם מספרי המטבעות של 10,5,1 סנט של O בהתאמה, אזי: $x \leq 2$, $y \leq 1$, $z \leq 4$.

<u>הוכחת טענה ראשית:</u> נסתכל על המולטי קבוצה S המתקבלת לאחר הבחירה האחרונה של האלגוריתם. ע"פ הטענה הנשמרת קיים פתרון אופטימלי O כך ש - $S \subseteq O$. מהגדרת האלגוריתם S הינו פתרון חוקי לבעיה, O הינו פתרון חוקי מינימלי אז $|S| \geq |O|$, ומכיוון ש - $S \subseteq O$ אזי $|S| = |O|$ ולכן $S=O$. בסה"כ הראינו כי האלגוריתם מחזיר את S שהינה פתרון אופטימלי לבעיה.

<u>הוכחת טענת עזר:</u> יהי O פתרון אופטימלי כלשהו ו – x,y,z מספר המטבעות של 10,5,1 סנט שלו בהתאמה. נניח בשלילה כי $z \geq 5$ אזי נוכל לקחת מטבע 1 של 5 סנט במקום 5 מטבעות של 1 סנט בסתירה למינימליות O. נניח בשלילה כי $y \geq 2$ אזי נוכל לקחת מטבע 1 של 10 סנט במקום 2 מטבעות של 5 סנט בסתירה למינימליות O. נניח בשלילה כי $x \geq 3$ אזי נוכל לקחת מטבע 1 של 25 סנט ומטבע 1 של 5 סנט במקום 3 מטבעות של 10 סנט בסתירה למינימליות O.

<u>הוכחת טענה נשמרת:</u> נוכיח באינדוקציה על i, כאשר i הינו שלב הבחירה של האלגוריתם.

*בסיס: עבור i=0, $S_0 = \emptyset$ ולכן מוכלת בכל פתרון אופטימלי לבעיה.

*הנחה: יהי i כלשהו ונניח כי S_{i-1} מוכלת ב – O פתרון אופטימלי כלשהו לבעיה.

*צעד: נסתכל על הבחירה ה – i של האלגוריתם ונסמנה a, $a \in \{1,5,10,25\}$. אם $S_{i-1} \cup \{a\} \subseteq O$, סיימנו. אחרת $S_{i-1} \cup \{a\} \nsubseteq O$.

נסמן את סכום איברי הקבוצה $O \backslash S_{i-1}$ ב – P, ונבחין כי במידה ובשלב ה – i האלגוריתם בחר במטבע של a סנט אזי P גדול שווה ל – a (נותרו לאלגוריתם לבחור עוד לפחות a סנט על מנת להגיע לפתרון חוקי) ו – P קטן שווה מגודל המטבע שבא לאחר a (האלגוריתם בוחר את המטבע הגדול ביותר שהוא יכול בכל שלב, לכן לא יתכן כי נותרו לו a סנט או יותר לבחור לפני השלב ה - i).

בנוסף, נסמן ב - x,y,z את מספרי המטבעות של 10,5,1 סנט של O בהתאמה.

<u>מקרה 1: a=1</u>

האלגוריתם בחר מטבע של 1 סנט, ולכן $1 \leq P \leq 4$. לא ניתן ליצור את P מקומבינציה של מטבעות 5,10,25 סנט ולכן בהכרח $S_{i-1} \cup \{a\} \subseteq O$. סתירה.

מקרה 2: a=5

האלגוריתם בחר מטבע של 5 סנט, ולכן $5 \leq P \leq 9$. לא ניתן ליצור את P מקומבינציה של מטבעות 10,25 סנט, $O \backslash S_{i-1}$ אינה מכילה מטבעות של 5 סנט (מכיוון ש - $S_{i-1} \subseteq O$ אולם $S_{i-1} \cup \{5\} \nsubseteq O$) ולכן $O \backslash S_{i-1}$ מכילה P מטבעות של 1 סנט ומכאן ש - $5 \leq P = z$ בסתירה לטענת העזר.

מקרה 3: a=10

האלגוריתם בחר מטבע של 10 סנט, ולכן $10 \leq P \leq 24$. לא ניתן ליצור את P מקומבינציה של מטבעות 25 סנט, $O \backslash S_{i-1}$ אינה מכילה מטבעות של 10 סנט (מכיוון ש - $S_{i-1} \subseteq O$ אולם $S_{i-1} \cup \{10\} \nsubseteq O$) ולכן $O \backslash S_{i-1}$ מכילה P סנט מקומבינציה של מטבעות 1,5 סנט. $10 \leq P$ לכן $2 \leq y$ או $5 \leq z$ בסתירה לטענת העזר.

מקרה 4: a=25

האלגוריתם בחר מטבע של 25 סנט, ולכן $25 \leq P$. $S_{i-1} \cup \{25\} \nsubseteq O$ אזי $O \backslash S_{i-1}$ מכילה רק מטבעות של 1,5,10 סנט. ע"פ טענת העזר $x \leq 2$, $y \leq 1$, $z \leq 4$, ומכך ש - $O \backslash S_{i-1}$ מכילה רק מטבעות של 1,5,10 סנט נקבל כי $P \leq 2 * 10 + 1 * 5 + 4 * 1 = 29$. בסה"כ קיבלנו כי $25 \leq P \leq 29$. קל לראות כי פריטה של P סנט בעזרת מטבעות של 1,5,10 סנט במקום מטבעות של 1,5,10,25 סנט מגדילה את מספר המטבעות בלפחות 2 מטבעות (25 בעזרת 10,10,5 במקום 25, והשאר בעזרת מטבעות של 1 סנט) בסתירה למינימליות O.

חישוב זמן ריצה:

אתחול האלגוריתם מתבצע בזמן של – O(1).

במהלך האלגוריתם אנו מבצעים O(n) איטרציות כשבכל אחת מתבצעות פעולות ב – O(1).

בסה"כ קיבלנו כי זמן הריצה של האלגוריתם הינו *O(1)+O(n)*O(1) = O(n)*.

שאלה 2

תיאור האלגוריתם:
1. מיין את סדרת הפרויקטים ע"פ הדד ליין מהקטן לגדול.
2. ניצור ערימת מקסימום ריקה של פרויקטים ע"פ הצעת מחיר, ורשימה מקושרת ריקה.
3. נכניס לערימת המקסימום את הפרויקט האחרון בסדרה (בעל הדד ליין הגדול ביותר).
4. נעבור על סדרת הפרויקטים הממוינת מהסוף (n-1) להתחלה (1): אם הדד ליין של הפרויקט אותו אנו
בודקים שווה לדד ליין של הפרויקט האחרון שהכנסנו לערימה, אז נכניסו לערימת המקסימום ונמשיך לפרויקט
הבא. אחרת, נשלוף מערימת המקסימום לראש הרשימה המקושרת (כל עוד יש בערימה איברים) מספר
פרויקטים כהפרש בין הדד ליין של הפרויקט האחרון שהכנסנו לערימה לפרויקט אותו אנו בודקים. לבסוף,
נכניס לערימה את הפרויקט אותו אנו בודקים.
5. כאשר הגענו לבדוק את הפרויקט הראשון בסדרה, נשלוף מערימת המקסימום לראש הרשימה
המקושרת (כל עוד יש בערימה איברים) מספר פרויקטים כגודל הדד ליין שלו.
6. נחזיר את הרשימה המקושרת.
הוכחת נכונות:
טענה ראשית: האלגוריתם החמדן שתיארנו מחזיר פתרון אופטימלי לבעיה.
טענה נשמרת: אם S_i הינה סדרה כלשהי המתקבלת לאחר הוספת האיבר ה – i לרשימה המקושרת
באלגוריתם, אזי S_i הינה סיפא של פתרון O אופטימלי כלשהו.
טענת עזר 1: אם S הינו פתרון של האלגוריתם אז S הינו פתרון חוקי.
טענת עזר 2: אם O הינו פתרון אופטימלי כלשהו ו – P פתרון חוקי כלשהו אז $length(P) \le length(O)$
טענת עזר 3: אם S הינו פתרון של האלגוריתם ו – P פתרון חוקי כלשהו אז $length(P) \le length(S)$
הוכחה טענה ראשית: נסתכל על הסדרה S המתקבלת לאחר הוספת האיבר האחרון לרשימה המקושרת.
ע"פ הטענה הנשמרת, S הינה סיפא של פתרון אופטימלי כלשהו O. ע"פ טענת עזר 3
$length(O) \le length(S)$. S היא סיפא של O ואורכה גדול שווה מאורכו ולכן S=O, ומכאן שהאלגוריתם
מחזיר פתרון אופטימלי לבעיה.
הוכחת טענה נשמרת: נוכיח באינדוקציה על i: *בסיס i=0: S_0 הינה סדרה ללא איברים ולכן הינה סיפא של כל
סדרה שקיימת, ובפרט סיפא של פתרון אופטימלי O כלשהו. *הנחה: יהי i ונניח כי S_{i-1} הינה סיפא של פתרון
אופטימלי O כלשהו. *צעד: נסתכל על הסדרה S_i שמתקבלת לאחר הוספת האיבר ה – i:
נסמן $S = p_{j_1}, p_{j_2}, \dots, p_{j_{l-i+1}}, \dots, p_{j_l}$ כפתרון שהאלגוריתם יחזיר. ע"פ שילוב של טענות עזר 1,2,3 נקבל כי
$length(S) = length(O) = l$. לכן, נוכל לסמן $O = p_{m_1}, p_{m_2}, \dots, p_{m_{l-i+1}}, \dots, p_{m_l}$.
מקרה 1 $(p_{j_{l-i+1}} = p_{m_{l-i+1}})$: (האיבר ה – i שנוסף ל – S שווה לאיבר ה – i מהסוף של O) סיימנו.
מקרה 2 $(p_{j_{l-i+1}} \neq p_{m_{l-i+1}})$: מקרה 2.1 - $p_{j_{l-i+1}}$ הינו איבר כלשהו ב – O:
$O = p_{m_1}, \dots, p_{m_k} = p_{j_{l-i+1}}, \dots, p_{m_{l-i+1}}, \dots, p_{m_l}$. נסתכל על $O' = p_{m_1}, \dots, p_{m_{l-i+1}}, \dots, p_{m_k} = p_{j_{l-i+1}}, \dots, p_{m_l}$.
נבחין כי 'O הינו פתרון חוקי (מכיוון שהפרויקט $p_{m_{l-i+1}}$ הוקדם לחודש הקודם ממקומו ב – O מה שלא יכול

לפגוע בחוקיות, ו - $p_{j_{l-i+1}}$ אוחר לחודש $l-i+1$ כפי שהופיע ב – S שהינו פתרון חוקי). בנוסף, הרווח של O'
זהה לרווח של O ולכן O' הינו פתרון אופטימלי, ומכאן ש - S_i הינה סיפא של פתרון אופטימלי כלשהו.
מקרה 2.2 ($p_{j_{l-i+1}}$ אינו איבר כלשהו ב – O): מקרה 2.2.1 ($m(p_{j_{l-i+1}}) = m(p_{m_{l-i+1}})$): אז הסדרה O'
הנוצרת ע"י החלפת $p_{m_{l-i+1}}$ ב - $p_{j_{l-i+1}}$ בסדרה O הינה פתרון אופטימלי לבעיה, וסיימנו.
מקרה 2.2.2 ($m(p_{j_{l-i+1}}) > m(p_{m_{l-i+1}})$): אז הסדרה O' הנוצרת ע"י החלפת $p_{m_{l-i+1}}$ ב - $p_{j_{l-i+1}}$ בסדרה O
הינה פתרון חוקי בעל רווח הגדול מהרווח של O בסתירה לאופטימליות O.
מקרה 2.2.3 ($m(p_{j_{l-i+1}}) < m(p_{m_{l-i+1}})$): ראשית נשים לב כי $d(p_{j_{l-i+1}}) < d(p_{m_{l-i+1}})$, אחרת ע"פ הגדרת
האלגוריתם $p_{m_{l-i+1}}$ היה נבחר לפני $p_{j_{l-i+1}}$ ב – S. בנוסף, מכיוון שנשלפים איברים מהערימה ע"פ הפרש
דד-ליינים, ניתן להבחין כי לכל $p_{j_t} \in \{p_{j_{l-i+2}}, \dots, p_{j_l}\}$ מתקיים $d(p_{j_t}) - d(p_{j_{l-i+1}}) \geq t-(l-i+1)$.
$d(p_{j_{l-i+1}}) < d(p_{m_{l-i+1}})$, וע"פ הנחת האינדוקציה ידוע כי לכל $t \in \{l-i+2, \dots, l\}$ מתקיים $d(p_{m_t}) = d(p_{j_t})$
ולכן $d(p_{m_t}) - d(p_{m_{l-i+1}}) > t-(l-i+1)$ לכל $t \in \{l-i+2, \dots, l\}$. O הינו פתרון חוקי אז $d(p_{m_{l-i+1}}) \geq l-i+1$
ובסה"כ קיבלנו כי $d(p_{m_t}) > t$ לכל $t \in \{l-i+2, \dots, l\}$. לכן, ניתן לראות כי ב – O ניתן לבצע כל פרויקט מבין
הפרויקטים $p_{m_{l-i+2}}, \dots, p_{m_l}$ חודש אחד מאוחר יותר, ומכאן ש - $O' = p_{m_1}, p_{m_2}, \dots, p_{m_{l-i+1}}, p_{j_{l-i+1}}, p_{m_{l-i+2}}, \dots, p_{m_l}$
הינו פתרון חוקי בסתירה למקסימליות אורך (או רווח) O.
<u>הוכחת טענת עזר 1</u>: יהי $S = p_{j_1}, \dots, p_{j_l}$ פתרון כלשהו של האלגוריתם. נניח בשלילה כי קיים p_{j_i} כך ש -
$d(j_i) < i$. ע"פ תיאור האלגוריתם, מספר האיברים ב – S שלפני p_{j_i} הינו כמספר המשיכות מתוך הערימה
ולכל היותר i-1 איברים בסתירה להנחה.
<u>הוכחת טענת עזר 2</u>: יהיו O פתרון אופטימלי ו – P פתרון חוקי כלשהו ונניח בשלילה כי $length(P) > length(O)$
הסדרה O' הנוצרת מהוספת האיבר ה – length(O)+1 של P בסוף הסדרה O בעלת רווח גדול יותר משל O,
בסתירה לאופטימליות O.
<u>הוכחת טענת עזר 3</u>: יהיו S פתרון של האלגוריתם ו – P פתרון חוקי כלשהו ונניח בשלילה כי
$length(P) > length(S)$. נסתכל מההתחלה על S ונמצא את האיבר הראשון p_i כך ש - $d(p_i) > i$.
(הדד ליין של האיבר האחרון ב – S הוא מקסימלי, אז אם לא קיים כזה זה אומר שקיים ב – P איבר בחודש
גדול מהדד ליין המקסימלי של המופע). מהגדרת האלגוריתם, כל הפרויקטים בעלי דד-לין גדול שווה i
היו בערימה לפני או בעת הוצאת p_i. אנו יודעים כי קיימים length(P) – i איברים כאלו לכל הפחות, מחוקיות P.
ומכאן, שגם לאחר הוצאת length(S) - i איברים מהערימה נותרו לכל הפחות (length(P) -i) - (length(S) – i),
כלומר לפחות איבר אחד. ומכיוון שהאלגוריתם מוציא מהערימה מספר פרויקטים כהפרש הדד-ליינים, והיו
לו איבר נוסף להוציא, זו סתירה לחוקיות האלגוריתם (בין p_i לאיבר שלפניו ב – S היה יכול להישלף איבר
כלשהו שהראינו שאכן היה בערימה, אך לא נשלף).
<u>ניתוח זמן ריצה</u>: מיון הפרויקטים מתבצע ב – O(nlogn). במהלך האלגוריתם אנו מבצעים לכל היותר n פעולות
הכנסה לערימה ולכל היותר n פעולות הוצאה מהערימה שגודלה לכל היותר n, מכאן שעלות ההכנסות הינה
O(nlogn) ועלות ההוצאות הינה O(nlogn). נשים לב כי הכנסה לרשימה מקושרת הינה O(1), אנו מבצעים l
פעולות כאלה ב – $O(l)$. $l \leq n$ אז זמן הריצה הכולל של האלגוריתם הינו *O(nlogn)*.

שאלה 3

סעיף א'

<u>תיאור האלגוריתם:</u>

בהינתן הצלע e=(v,u), נבצע סריקת DFS בגרף T מקודקוד v, אשר בסופה לכל קודקוד ב – V יהיה שדה π אשר בו מופיע מי הקודקוד הקודם במסלול הקצר ביותר מ – v אליו.

נסרוק מקודקוד u חזרה לקודקוד v ע"פ השדה π, ובכל צלע שנעבור נשווה את משקלה עם משקל e. אם משקלה גדול ממשקל e נחזיר false, אחרת נמשיך לצלע הבאה.

במידה והגענו לקודקוד v, נחזיר true.

<u>הוכחת נכונות:</u>

<u>טענה ראשית:</u> האלגוריתם מחזיר האם T הוא MST של הגרף G'.

<u>טענת עזר:</u> T הוא MST של G' אמ"ם לכל $e' \neq e$ השייכת למעגל שנסגר בעזרת e וצלעות T מתקיים $w(e') \leq w(e)$.

<u>הוכחת טענה ראשית:</u> האלגוריתם עובר על כל הצלעות ב – T שביחד עם e **יוצרות מעגל, ומשווה בין משקלן לבין משקל** e. במידה וקיימת צלע כזו בעלת משקל הגדול ממשקל e אז ע"פ טענת העזר T אינו MST של G' **ואכן** האלגוריתם מחזיר false. במידה ולא קיימת צלע כזו, אז שוב ע"פ טענת העזר T הינו MST של G' **ואכן** האלגוריתם מחזיר true.

<u>הוכחת טענת עזר:</u> ⇐ נניח כי T הוא MST של G'. נניח בשלילה שקיימת e' כך $w(e') > w(e)$. ע"פ טענה שהוכחנו בכיתה $T' = \{V, E_T \setminus \{e\} \cup \{e'\}\}$ הינו עץ פורש של G, ולכן עץ פורש גם של G', בעל משקל הקטן ממשקל T בסתירה לכך ש – T הינו MST של G'.

⇒ נניח בשלילה כי T אינו MST של G'. ניתן להבחין כי T הינו עץ פורש של G', אז לפי ההנחה קיים T' שהינו גם כן עץ פורש של G' אך משקלו קטן ממש ממשקל T.

מקרה 1 ($e \notin E_{T'}$): אז T' הינו עץ פורש של G שמשקלו קטן ממש ממשקל T בסתירה למינימליות T.

מקרה 2 ($e \in E_{T'}$): נניח בשלילה כי לכל $e' \neq e$ השייכת למעגל שנסגר בעזרת e וצלעות T מתקיים $w(e') \leq w(e)$.

מקרה 2.1 (קיים $e' \neq e$ כך ש - $w(e') < w(e)$): אזי כל פי משפט שלמדנו בכיתה במידה ונסיר מ – T' את e ונוסיף את e' נקבל עץ פורש של G' בעל משקל הקטן ממשקל T' בסתירה למינימליות T'.

מקרה 2.2 (לכל $e' \neq e$ מתקיים $w(e') = w(e)$): נסתכל על העץ הפורש שמתקבל על ידי הסרת e והוספת e' כלשהי. קיבלנו עץ שמשקלו זהה למשקל T' ולכן משקלו קטן ממש ממשקל T. מצד שני, עץ זה פורש את G וממינימליות T משקלו גדול שווה מ – T. סתירה.

<u>חישוב זמן ריצה:</u> סריקת ה – DFS תעשה ב O(|V|+|E|), והסריקה שלאחריה תעשה ב O(|E|), ומכאן שזמן הריצה הכולל של האלגוריתם הינו O(|V|+|E|).

סעיף ב'

<u>תיאור האלגוריתם:</u>

ניצור גרף חדש $G' = (V\setminus U, E\setminus E_U)$ ונפעיל עליו את אלגוריתם קרוסקל שלמדנו בכיתה למציאת MST של G', נסמן את העץ שקיבלנו ב – T'. כעת, נשכפל את U ל – U' ונמיין את צלעות E ע"פ משקלן. כל עוד $U' \neq \emptyset$ נבחר את הצלע בעלת המשקל הנמוך ביותר ב – E. במידה וזו צלע מקודקוד u השייך ל – U' לקודקוד השייך ל – V\U (נבדוק בעזרת מערך בינארי) נוסיפה לקבוצת E' ונסיר את u מ- U' (במידה וסיימנו לעבור על כל צלעות E ונותרו קודקודים ב – U' האלגוריתם יחזיר כי לא קיים $\mathrm{MST_U}$).

בסוף הלולאה נחזיר את הגרף $\mathrm{T} = (\mathrm{V}, \mathrm{E_{T'}} \cup \mathrm{E'})$.

<u>הוכחת נכונות:</u>

<u>טענה ראשית:</u> אם קיים $\mathrm{MST_U}$ עבור מופע כלשהו אז האלגוריתם מחזיר אחד כזה, אחרת יחזיר כי אין.

<u>טענת עזר:</u> קיים $\mathrm{MST_U}$ אמ"ם לכל $\mathrm{u} \in \mathrm{U}$ קיים $\mathrm{v} \in \mathrm{V} \setminus \mathrm{U}$ כך ש - $(u, v) \in \mathrm{E}$.

<u>הוכחת טענה ראשית:</u> במידה ולא קיים $\mathrm{MST_U}$ אז ע"פ טענת העזר קיים $\mathrm{u} \in \mathrm{U}$ כך שלכל $(u, v) \in \mathrm{E}$ מתקיים $\mathrm{v} \in \mathrm{U}$. במקרה זה אכן האלגוריתם יחזיר כי לא קיים $\mathrm{MST_U}$.

במידה וכן קיים $\mathrm{MST_U}$, נסתכל על הגרף T שמחזיר האלגוריתם. ראשית, נראה כי T אכן עץ פורש של G. $|\mathrm{E_{T'}} \cup \mathrm{E'}| = |\mathrm{E_{T'}}| + |\mathrm{E'}| = |V\setminus U| - 1 + |U| = |V| - 1$. בנוסף, T' הינו קשיר וכל צלע ב E' מחברת קודקוד כלשהו מ – U לרכיב קשירות זה, ולכן T קשיר. $|\mathrm{E_{T'}} \cup \mathrm{E'}| = |V| - 1$ וגם T קשיר אזי T הינו עץ פורש של G, ומעצם בנייתו ניתן לראות כי כל קודקוד השייך ל – U הינו עלה ב – T. נניח בשלילה כי קיים $\mathrm{MST_U}$ של G נסמנו P כך שמשקל P קטן ממש ממשקל T. נתבונן בעץ P' המתקבל מהסרת קודקודי U מ – P ומכל הצלעות החלות עליהם בו. ניתן לראות כי שמרנו על קשירות במהלך זה והסרנו מספר צלעות כמספר הקודקודים ולכן העץ P' שהתקבל הינו עץ פורש עבור G'. ממינימליות T' משקלו של P' גדול שווה ממשקלו של T'. בנוסף, ע"פ הגדרת האלגוריתם משקל הצלעות שהסרנו מ – P גדול שווה ממשקל הצלעות של E'. משקלו של P שווה למשקל P' ועוד משקל הצלעות שהסרנו, ולפי מה שראינו משקל זה גדול שווה ממשקל T בסתירה להנחה כי משקל P קטן ממש ממשקל T.

<u>הוכחת טענת עזר:</u> ⇐ נניח כי קיים $\mathrm{MST_U}$. נסתכל על הצלעות בעץ זה החלות על קודקודי U, ונבחין כי כל צלע כזו הינה צלע מקודקוד מ – U לקודקוד מ – V\U (צלע מ – U ל – U תגרום לכך שקודקוד כלשהו ב – U אינו עלה). אז לכל $\mathrm{u} \in \mathrm{U}$ קיים $\mathrm{v} \in \mathrm{V} \setminus \mathrm{U}$ כך ש - $(u, v) \in \mathrm{E}$.

⇒ נניח כי לכל $\mathrm{u} \in \mathrm{U}$ קיים $\mathrm{v} \in \mathrm{V} \setminus \mathrm{U}$ כך ש - $(u, v) \in \mathrm{E}$. נחבר כל קודקוד מ – U לקודקוד מ – $V\setminus U$ בעזרת צלעות אלה, וניצור עץ הפורש את קודקודי V\U. ניתן להבחין כי קיבלנו עץ הפורש את G כך שכל קודקוד מ – U הינו עלה. אם קיים עץ כזה, אז קיים עץ כזה בעל משקל מינימלי הלא הוא $\mathrm{MST_U}$.

<u>ניתוח זמן ריצה:</u>

בניית G' תעשה ב- O(|V|+|E|), ביצוע אלגוריתם קרוסקל יעשה ב- O(|E|*log(|V|)). שכפול U יעשה ב – O(|V|). מיון צלעות E יעשה בזמן O(|E|*log|E|) ומעבר עליהן בזמן של O(|E|). הגרף קשיר לכן |V|=O(|E|), ובסה"כ קיבלנו כי זמן הריצה הכולל הינו: O(|E|*log(|E|)).

שאלה 4