אלגוריתמים – ממ״ן 13

שאלה 1

נוכיח את הטענה עבור כל מסלול המתחיל ב-s (ומסתיים לאו דווקא ב-t), באינדוקציה על k, מספר הקשתות במסלול. בסיס (k=1): המסלול מכיל קשת אחת שימושית (s, u), ולכן ע״פ הגדרה קיים מסלול קצר ביותר מ-s ל-u שמסתיים בקשת הזאת, כלומר הקשת עצמה מהווה מסלול קצר ביותר מ-s ל-u. נניח נכונות עבור מסלולים באורך k, ויהי p מסלול עם k+1 קשתות מ-s ל-u, כאשר הקשת האחרונה במסלול היא e=(v, u). עד v יש במסלול k קשתות וכולן שימושיות ולכן ע״פ הנחת האינדוקציה הרישא של המסלול עד v היא מסלול קצר ביותר מ-s ל-v שמשקלה לכן $\delta(v)$. לכן משקל המסלול p הוא $\delta(v)+w(e)$. מאחר ש-e קשת שימושית אז ע״פ הגדרה יש מסלול קצר ביותר מ-s ל-u ש-e קשת אחרונה בו. נקרא למסלול זה p'. משקלו של p' הוא $\delta(u)$ והוא בהכרח שווה ל-$\delta(v)+w(e)$, כלומר למשקלו של p. לכן גם משקלו של p הוא $\delta(u)$ והוא מסלול קצר ביותר מ-s ל-u.

ב. יהי p מסלול בין s לצומת כלשהו u ובו קשת לא שימושית e=(w, v). הרישא של המסלול מ-s ל-v בהכרח אינה מסלול קצר ביותר מ-s ל-v (אחרת e הייתה שימושית) ולכן קיים מסלול p' מ-s ל-v שהוא קצר יותר מהרישא של p עד v. לכן אפשר להחליף ב-p את הרישא עד v במסלול p' ולקבל מסלול מ-s ל-u שהוא קצר יותר מ-p ולכן p אינו מסלול קצר ביותר. הראינו זאת לכל צומת u ובפרט זה נכון עבור u=t.

ג. יהי p מסלול שני קצר ביותר מ-s ל-t. ברור שיש בו קשת לא שימושית, אחרת ע״פ סעיף א׳ הוא היה מסלול קצר ביותר. תהי e=(u, v) הקשת הלא-שימושית הראשונה ב-p. יהי המסלול p_1 הרישא של p מ-s עד u. p_1 מכיל רק קשתות שימושיות ולכן זהו מסלול קצר ביותר מ-s ל-u ע״פ סעיף א׳. כעת נניח שבהמשך המסלול p יש עוד קשת לא שימושית ותהי e'=(u', v') הקשת הלא שימושית הראשונה בהמשך המסלול. כלומר, מ-v ל-u' כל הקשתות הן שימושיות. נסמן ב- p_2 את קטע

המסלול מ-v ל-'v. p_2 בהכרח אינו מסלול קצר ביותר מ-v ל-'v, משום שאם היה קצר ביותר ניתן היה לקחת מסלול קצר ביותר מ-s ל-v, לשרשר לו את p_2 ולקבל מסלול קצר ביותר מ-s ל-'v המכיל קשת לא שימושית, בסתירה לסעיף ב׳ (שימו לב שלא ניתן להשתמש ישירות בסעיף ב׳ כדי להראות ש-p_2 אינו קצר ביותר כי סעיף ב׳ נוסח עבור מסלולים המתחילים ב-s). מאחר ש-p_2 אינו מסלול קצר ביותר מ-v ל-'v, אז קיים מסלול p_3 מ-v ל-'v שהוא קצר יותר מ-p_2. נחליף ב-p את הקטע p_2 ב-p_3 ונקבל מסלול חדש מ-s ל-t. מסלול זה קצר יותר מ-p, אבל הוא מכיל קשת לא שימושית ולכן ע״פ סעיף ב׳ אינו מסלול קצר ביותר מ-s ל-t. לכן p אינו מסלול שני קצר ביותר מ-s ל-t בסתירה לנתון.

עד כה הראינו שיש ב-p רק קשת שימושית אחת, ושעד אליה זהו קטע מסלול קצר ביותר. נותר עוד להראות שהסיפא של p מ-v עד t מהווה מסלול קצר ביותר מ-v ל-t (לא ניתן להשתמש בסעיף א׳ כי זהו אינו מסלול שמתחיל ב-s). נקרא לסיפא זו p_4. נניח שקיים מ-v ל-t מסלול p_5 שהוא קצר יותר מ-p_4. בדומה לטיעון בפסקה הקודמת, ניתן להחליף ב-p את הקטע p_4 בקטע p_5, ולקבל מסלול חדש מ-s ל-t. מסלול זה קצר יותר מ-p, אבל הוא מכיל קשת לא שימושית ולכן ע״פ סעיף ב׳ אינו מסלול קצר ביותר מ-s ל-t. לכן p אינו מסלול שני קצר ביותר מ-s ל-t בסתירה לנתון.

ד. נתאר אלגוריתם שמשתמש באלגוריתם של דייקסטרא כקופסה שחורה: על ידי שתי הרצות של דייקסטרא, פעם על הגרף המקורי מ-s ופעם על הגרף המוחלף מ-t אפשר למצוא לכל צומת v את מרחקו מ-s ואת מרחק t ממנו. נסמן את הערכים האלו ב-$\delta_s(v)$ וב-$\delta_t(v)$. כעת, קשת e=(u, v) היא שימושית אםם מתקיים $\delta_s(u)+w(e)=\delta_s(v)$ (וודאו כי הנכם יודעים להוכיח זאת). עתה, נעבור על כל קשת e=(u, v) בגרף, נבדוק (ב-O(1)) אם היא מקיימת את התנאי שלעיל, ואם היא אינה מקיימת את התנאי נחשב את הערך $\delta_s(u)+\delta_t(v)+w(e)$, ונשמור תוך כדי המעבר על כל הקשתות את הערך המינימלי ביותר שמצאנו. זהו משקלו של מסלול שני קצר ביותר. העלות היא $O(|E|+|V|\log|V|)$. הקשתות שעבורן חישבנו את הערך הזה הן כל הקשתות הלא-שימושיות.

ברור שכל אחד מהערכים שחושבו, ובפרט הערך המינימלי מביניהם, הוא משקל של איזשהו מסלול מ-s ל-t. מכך שחישבנו את הערך הזה רק עבור קשתות שאינן מקיימות את התנאי (כלומר, קשתות לא שימושיות), ברור שאף אחד מהערכים שחושבו אינו משקל מסלול קצר ביותר מ-s ל-t, ולכן הערך המינימלי שחושב הוא לכל היותר משקלו של מסלול שני קצר ביותר מ-s ל-t. מצד שני, ע"פ סעיף ג' עבור מסלול שני קצר ביותר p קיימת קשת e=(u, v) לא שימושית כך שהערך $\delta_s(u)+\delta_t(v)+w(e)$ הוא משקל המסלול p. לכן משקלו של מסלול שני קצר ביותר הוא אחד הערכים שחושבו. לכן מובטח שהערך המינימלי מבין כל הערכים שחושבו הוא אכן משקלו של מסלול שני קצר ביותר מ-s ל-t.

שאלה 2

א.

ראשית כל, לשם הנוחות נסמן :

$$f_i g_j = f_i(x)g_j(x)$$

$$\alpha_{ii} = f_i g_i$$

$$\gamma_{ij} = \left(f_i + f_j\right)\left(g_i + g_j\right) = \gamma_{ji}$$

כמו כן, נכנה בשם "הדרגה של הפולינום" את מספר המקדמים של הפולינום. בשאלה מבקשים להתייחס לפולינום ממעלה שהיא חזקה של שלוש\ארבע, אך החלוקה טבעית ביותר כאשר מספר המקדמים הוא חזקה של שלוש\ארבע, וזה גם המצב שמניחים בספר.

האלגוריתם מקבל שני פולינומים שמספר המקדמים בהם, n, הוא חזקה של 3, ומחלק אותם לשלושה חלקים באופן הבא : החלק הראשון : פולינום f_1 עם $n/3$ מקדמים המייצג את $n/3$ המקדמים של החזקות הגבוהות, פולינום f_2 עם $n/3$ מקדמים המייצג את $n/3$ המקדמים הבאים, וכן f_3 המייצג את המקדמים של החזקות הנמוכות. f ו- g מיוצגים בצורה זו כך :

$$f(x) = f_1 x^{2n/3} + f_2 x^{n/3} + f_3$$
$$g(x) = g_1 x^{2n/3} + g_2 x^{n/3} + g_3$$

נכפול ביטויים אלו ונקבל :

$$\begin{aligned} f(x)g(x) = {} & f_1 g_1 x^{4n/3} + f_1 g_2 x^{n} + f_1 g_3 x^{2n/3} \\ & + f_2 g_1 x^{n} + f_2 g_2 x^{2n/3} + f_2 g_3 x^{n/3} \\ & + f_3 g_1 x^{2n/3} + f_3 g_2 x^{n/3} + f_3 g_3 \end{aligned}$$

$$f(x)g(x) = x^{4n/3} f_1 g_1 + x^{n}\left(f_1 g_2 + f_2 g_1\right) + x^{2n/3}\left(f_1 g_3 + f_2 g_2 + f_3 g_1\right) + x^{n/3}\left(f_2 g_3 + f_3 g_2\right) + f_3 g_3$$

נשתמש בעובדה שמצויינת במדריך הלמידה : $f_i g_j + f_j g_i = \gamma_{ij} - \alpha_{ii} - \alpha_{jj}$ ונקבל :

$$f(x)g(x) = x^{4n/3}\alpha_{11} + x^{n}\left(\gamma_{12} - \alpha_{11} - \alpha_{22}\right) + x^{2n/3}\left(\gamma_{13} - \alpha_{11} - \alpha_{33} + \alpha_{22}\right) + x^{n/3}\left(\gamma_{23} - \alpha_{22} - \alpha_{33}\right) + \alpha_{33}$$

כלומר נחשב את שש המכפלות : $\alpha_{11},\alpha_{22},\alpha_{33},\gamma_{12},\gamma_{13},\gamma_{23}$, שהן מכפלות בין פולינומים בעלי $n/3$ מקדמים, באופן רקורסיבי. לאחר מכן נחשב את המקדמים על ידי חיבור וחיסור של פולינומים, ונציב את המקדמים שקיבלנו בפולינום התוצאה, באותו אופן בו חילקנו את הפולינום (ראה את הפסאודו קוד של אלגוריתם 5.3 במדריך - הפעולה מאוד דומה, אך הרבה יותר מסובך לכתוב אותה ולכן היא לא מצורפת). כשנגיע לכפל של שני פולינומים ממעלה אחת, נכפול אותם כמספרים ממשיים.

<u>הוכחת נכונות:</u>

נוכיח באינדוקציה שהאלגוריתם מחשב את מכפלת הפולינומים. עבור $n=1$ האלגוריתם מחשב כפל של מספרים ממשיים, ולכן הוא נכון. נניח שהאלגוריתם מחשב נכון את הכפל עבור פולינומים עם n מקדמים, ונרצה להוכיח שהוא מחשב את הכפל עבור פולינומים עם $3n$ מקדמים.

נתבונן בפעולה של האלגוריתם על שני פולינומים שמספר המקדמים של הגדול מבינהם הוא $3n$.

האלגוריתם משלים את מספר המקדמים של הפולינום השני ל- $3n$ באמצעות ריפוד באפסים.

האלגוריתם מחלק את הפולינומים לשלושה חלקים כל אחד, ומחשב נכון את ששת המכפלות של פולינומים בעלי n מקדמים, לפי הנחת האינדוקציה. על פי הפיתוח לעיל, הצירוף של התוצאות לפולינום המכפלה הוא נכון.

<u>ניתוח סיבוכיות:</u>

עבור קלט בגודל n (מספר המקדמים), האלגוריתם קורא לעצמו שש פעמים עבור קלט בגודל $n/3$.

האלגוריתם צריך לעבד את התוצאות, לחשב את הסכומים וההפרשים, ולהציב את התוצאות בפולינום המכפלה. כל אחד משלבים אלו לוקח $O(n)$, ולכן בסך הכל נדרש זמן $O(n)$.

לפיכך, נוסחת הנסיגה המתאימה היא :

$$T(n)=6T(n/3)+O(n)$$

לפי שיטת האב, מקרה 1,

$$T(n)=O\left(n^{\log_3 6}\right)$$

שזה בערך $O(n^{1.63})$.

ב. האלגוריתם מקבל שני פולינומים שמספר המקדמים בהם הוא חזקה של 4, ומחלק אותם לשלושה חלקים באופן הבא: החלק הראשון: פולינום f_1 בעל $n/4$ מקדמים המייצג את $n/4$ המקדמים של החזקות הגבוהות, פולינום f_2 המייצג את $n/4$ המקדמים הבאים, כנ"ל f_3, וכן f_4 המייצג את $n/4$ המקדמים של החזקות הנמוכות. f ו- g מיוצגים בצורה זו כך:

$$f(x) = f_1 x^{3n/4} + f_2 x^{2n/4} + f_3 x^{n/4} + f_4$$
$$g(x) = g_1 x^{3n/4} + g_2 x^{2n/4} + g_3 x^{n/4} + g_4$$

נכפול ביטויים אלו ונקבל:

$$\begin{aligned} f(x)g(x) = {} & f_1 g_1 x^{6n/4} + f_1 g_2 x^{5n/4} + f_1 g_3 x^{n} + f_1 g_4 x^{3n/4} \\ & + f_2 g_1 x^{5n/4} + f_2 g_2 x^{n} + f_2 g_3 x^{3n/4} + f_2 g_4 x^{2n/4} \\ & + f_3 g_1 x^{n} + f_3 g_2 x^{3n/4} + f_3 g_3 x^{2n/4} + f_3 g_4 x^{n/4} \\ & + f_4 g_1 x^{3n/4} + f_4 g_2 x^{2n/4} + f_4 g_3 x^{n/4} + f_4 g_4 \\ f(x)g(x) = {} & x^{6n/4} f_1 g_1 + x^{5n/4}\left(f_1 g_2 + f_2 g_1\right) + x^{n}\left(f_1 g_3 + f_2 g_2 + f_3 g_1\right) \\ & + x^{3n/4}\left(f_1 g_4 + f_2 g_3 + f_3 g_2 + f_4 g_1\right) + x^{2n/4}\left(f_2 g_4 + f_3 g_3 + f_4 g_2\right) + x^{n/4}\left(f_3 g_4 + f_4 g_3\right) + f_4 g_4 \end{aligned}$$

נשתמש בעובדה שמצויינת במדריך הלמידה: $f_i g_j + f_j g_i = \gamma_{ij} - \alpha_{ii} - \alpha_{jj}$ ונקבל:

$$\begin{aligned} f(x)g(x) = {} & x^{6n/4}\alpha_{11} + x^{5n/4}\left(\gamma_{12} - \alpha_{11} - \alpha_{22}\right) + x^{n}\left(\gamma_{13} + \alpha_{22} - \alpha_{11} - \alpha_{33}\right) \\ & + x^{3n/4}\left(\gamma_{14} - \alpha_{11} - \alpha_{44} + \gamma_{23} - \alpha_{22} - \alpha_{33}\right) + x^{2n/4}\left(\gamma_{24} - \alpha_{22} - \alpha_{44} + \alpha_{33}\right) \\ & + x^{n/4}\left(\gamma_{34} - \alpha_{33} - \alpha_{44}\right) + \alpha_{44} \end{aligned}$$

אם נסמן

$$\phi = \left(f_1 + f_2 + f_3 + f_4\right)\left(g_1 + g_2 + g_3 + g_4\right) = f_1 g_1 + f_1 g_2 + f_1 g_3 + f_1 g_4 + f_2 g_1 + \ldots + f_4 g_4.$$

אז במקום לחשב את γ_{14}, γ_{23}, מספיק לחשב רק את: $\gamma_{14} + \gamma_{23} = \phi - \gamma_{12} - \gamma_{13} - \gamma_{24} - \gamma_{34}$.

כלומר נחשב את תשע המכפלות: $\alpha_{11},\alpha_{22},\alpha_{33},\alpha_{44},\gamma_{12},\gamma_{13},\gamma_{24},\gamma_{34},\phi$, שהן מכפלות בין פולינומים עם $n/4$ מקדמים, באופן רקורסיבי. לאחר מכן נחשב את המקדמים על ידי חיבור וחיסור של פולינומים, ונציב את המקדמים שקיבלנו בפולינום התוצאה, באותו אופן בו חילקנו את הפולינום (ראה את הפסיידו קוד של אלגוריתם 5.3 במדריך - הפעולה מאוד דומה, אך הרבה יותר מסובך לכתוב אותה ולכן היא לא מצורפת). כשנגיע לכפל של שני פולינומים ממעלה אחת, נכפול אותם כמספרים ממשיים.

הוכחת נכונות:

ההוכחה שקולה למקרה של החזקות של 3, רק שהפעם אנו עושים את האינדוקציה על חזקות של 4, לכן יש להחליף כל מופע של הספרה 3 בספרה 4. כאן מחשבים תשעה תת בעיות בגודל n כדי להגיע לפתרון הבעיה בגודל $4n$.

ניתוח סיבוכיות:

עבור קלט בגודל n (מספר המקדמים), האלגוריתם קורא לעצמו תשעה פעמים עבור קלט בגודל $n/4$. האלגוריתם צריך לעבד את התוצאות, לחשב את הסכומים וההפרשים, ולהציב את התוצאות בפולינום המכפלה. כל אחד משלבים אלו לוקח $O(n)$, ולכן בסך הכל נדרש זמן $O(n)$.

לפיכך, נוסחת הנסיגה המתאימה היא:

$$T(n) = 9T(n/4) + O(n)$$

לפי שיטת האב, מקרה 1, מקבלים ש-

$$T(n) = O\left(n^{\log_4 9}\right) = O\left(n^{\frac{\lg 9}{\lg 4}}\right) = O\left(n^{\frac{2\lg 3}{2}}\right) = O\left(n^{\lg 3}\right)$$

שזה בערך $O(n^{1.59})$.

שאלה 3

א.

אם $n=1$, יש רק שורש יחידה אחד, והוא 1.

נניח ש- $n>1$. שורשי היחידה מסדר n הם הערכים $e^{2\pi ik/n}$, כאשר $k=0,\dots,n-1$. כלומר:

$1,e^{2\pi i/n},e^{4\pi i/n},e^{6\pi i/n},\dots,e^{2\pi i(n-1)/n}$.

ניתן לרשום זאת גם כך:

$$\sum_{k=0}^{n-1}\left(e^{2\pi i/n}\right)^k=\frac{\left(e^{2\pi i/n}\right)^n-1}{e^{2\pi i/n}-1}=\frac{e^{2\pi i}-1}{e^{2\pi i/n}-1}$$

(המעבר הראשון הוא סכום של סדרה הנדסית).

$e^{2\pi i}=\cos 2\pi+i\sin 2\pi=1$, ו- $e^{2\pi i/n}\neq 1$ כי $n\neq 1$ טבעי, ולכן מתקבל:

$$\frac{e^{2\pi i}-1}{e^{2\pi i/n}-1}=\frac{1-1}{e^{2\pi i/n}-1}=0$$

לפיכך, סכום שורשי היחידה מסדר n הוא 1 אם $n=1$, ו-0 אחרת.

ב.

$$1\cdot e^{2\pi i/n}\cdot e^{4\pi i/n}\cdot e^{6\pi i/n}\cdot e^{2\pi i(n-1)/n}=\exp\left(\frac{2\pi i}{n}\sum_{k=1}^{n-1}k\right)\overset{(*)}{=}\exp\left(\frac{2\pi i}{n}\frac{n(n-1)}{2}\right)=e^{\pi i(n-1)}=\left(e^{\pi i}\right)^{n-1}$$

(*) סכום סדרה חשבונית

מתקיים $e^{\pi i}=\cos\pi+i\sin\pi=-1$, ולכן מכפלת שורשי היחידה מסדר n היא: $(-1)^{n-1}$.

אם n זוגי, המכפלה שווה ל- -1, ואם n אי-זוגי, המכפלה שווה ל-1.

שאלה 4

א. אם נסמן $A=\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ אז:

$$A^2=\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} a & b \\ c & d \end{pmatrix}=\begin{pmatrix} a^2+bc & ab+bd \\ ac+cd & bc+d^2 \end{pmatrix}=\begin{pmatrix} a^2+bc & b(a+d) \\ c(a+d) & bc+d^2 \end{pmatrix}$$

לכן על מנת לחשב את A^2, מספיק לחשב את חמש המכפלות הבאות:

$$P_1=a^2$$
$$P_2=b(a+d)$$
$$P_3=c(a+d)$$
$$P_4=d^2$$
$$P_5=bc$$

ואז:

$$A^2=\begin{pmatrix} P_1+P_5 & P_2 \\ P_3 & P_4+P_5 \end{pmatrix}$$

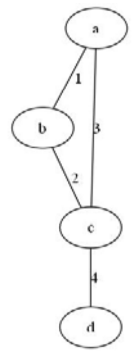
ב. הטענה **לא** נכונה. הסיבה לכך נעוצה בעובדה שבאלגוריתם שהפרופסור מציע, צריך לחשב את חמשת תת הבעיות באופן רקורסיבי. אבל לא כל המכפלות של מטריצות מסדר $\frac{n}{2}$ שאלגוריתם יצטרך לחשב הן מכפלה של מטריצה בעצמה: החישובים של P_2,P_3,P_5 הם חישובי כפל בין שתי מטריצות, שייתכן שהן שונות, ולכן לא ניתן להפעיל את אלגוריתם ההעלאה בריבוע באופן רקורסיבי עליהן. כדי שהאלגוריתם יעבוד צריך הפרופסור למצוא דרך לכפול שתי מטריצות שונות מסדר $\frac{n}{2}$ בעזרת 5 תת-בעיות. עוד סיבה לכישלון של האלגוריתם של הפרופסור היא שכפל מטריצות הוא לא קומוטטיבי, אך משתמשים בתכונה זו בסעיף א'.

הערה: למעשה אלגוריתם כזה היה גורר אלגוריתם שזמן הריצה שלו הוא $O(n^{\lg 5})$. הסיבה היא שלא קשה להראות כי כפל מטריצה אינו קשה יותר מהעלאה בריבוע: אלגוריתם יעיל לחישוב ריבוע של

מטריצה גורר אלגוריתם באותה סיבוכיות לכפל מטריצות (נסו להוכיח זאת). אלגוריתם שזו סיבוכיות זמן הריצה שלו לבעיה אינו ידוע. מתן אלגוריתם כזה הוא בעיה פתוחה מרכזית.

שאלה 5 (שאלה 4.9 מספר הלימוד)

א. הטענה לא נכונה. נסמן ב- G את הגרף:



קל לאמת שזהו גרף קשיר, עם משקלים חיוביים השונים זה מזה. על ידי הרצת האלגוריתם של קרוקסל קל לראות שהקשתות $\{b,c\},\{a,b\},\{c,d\}$ יוצרות עץ פורש מינימלי שמשקלו $1+2+4=7$.

ניתן לראות שכל עץ פורש של G חייב לכלול את הקשת $\{c,d\}$, ומכיוון שמשקלה הכי גדול מבין משקלי הקשתות של הגרף, זוהי קשת צוואר הבקבוק בכל עץ פורש של G.

נתבונן בעץ הפורש הנוצר על ידי הקשתות $\{a,c\},\{b,c\},\{c,d\}$. לפי האמור לעיל, משקל קשת צוואר הבקבוק הוא 4, ואין עץ פורש של G שמשקל קשת צוואר הבקבוק שלו קטן יותר. לפיכך, עץ זה הוא עץ פורש בעל צוואר בקבוק מינימלי. עם זאת, משקל העץ הוא $3+2+4=10$, ולכן הוא לא עץ פורש מינימלי של G.

ב. הטענה נכונה. הוכחה:

נניח בשלילה ש- T עץ פורש מינימלי של G, ושהוא לא עץ פורש בעל צוואר בקבוק מינימלי. יהי T' עץ פורש בעל צוואר בקבוק מינימלי כלשהו של G.

מכיוון ש- T הוא לא עץ פורש בעל צוואר בקבוק מינימלי, קשת צוואר הבקבוק של T, e, גדולה מכל הקשתות של העץ T'. הוכחה: נניח שקיימת קשת e' ב- T כך ש- $w(e) \le w(e')$. לפי הגדרת קשר צוואר הבקבוק, קשת צוואר הבקבוק של T', b, מקיימת: $w(b) \ge w(e')$ (ייתכן שזו e' בעצמה). לכן $w(b) \ge w(e)$, אך זוהי סתירה לכך ש-T' עץ פורש בעל צוואר בקבוק מינימלי.

אם נוסיף את הקשת e ל-T' יווצר מעגל c. המעגל c הוא מעגל בגרף G, המורכב מהקשת e, ומקשתות נוספות שכולן בגרף T'. לכן משקל הקשת e גדול ממשקליהן של כל שאר הקשתות במעגל. לפי תכונת המעגל (משפט 4.20), e לא נמצאת באף עץ פורש מינימלי של G, וזאת סתירה לכך ש- T עץ פורש מינימלי.