פרק 6

דוגמא ליישום של הגרפים : עץ פורש מינימלי

בתת פרק זה נתוודע אל בעיה מהעולם הממשי, בתחום של חקר ביצועים, ונציג לה פתרונות שונים.להלן ניסוח והצגת הבעיה ופתרונות (אלגוריתמים) שונים.

6.1 הצגת הבעיה

נתון גרף בלתי **מכוון קשיר** $G=(V,E)$ כאשר V מבטא קבוצת קודקודים ו E קבוצת קשתות.פונקצית משקל $w:E\rightarrow R$ קובעת משקל לכל קשת בגרף G. וכל קשת $(u,v)\in E$ היא בעלת משקל $w(u,v)$.משקל לכל קשת (u,v) יכול לייצג מרחק בין שני הקודקודים u ו v או ממוצע מספר המכוניות העוברות בין שני ערים המיוצגים באמצעות שני הקודקודים בגרף או עלות לסלילת כביש בין שני ערים המיוצגות באמצעות שני קודקודי הגרף וכדומה.

רשת זו יכולה לייצג מערכות קשר ותחבורה, כגון רשתות כבישים,מערכת כבלי טלפון, מערכות כבלים לחיבור כל תחנות טלוויזיה לרשת מסוימת, רשת תקשורת מחשבים וסלילת דרכים בין נקודות יישום שונות (כדי לחברם למערכת תחבורתית).

הבעיה היא:נתון גרף $G=(V,E)$. עלינו למצוא תת גרף $T=(V,E_T)$ ללא מעגלים (בקיצור עץ) של קשתות המחברות את כל קודקודי הגרף הנתון G ואשר משקלה הכולל $w(E_T)=\sum_{(u,v)\in E_T} w(u,v)$ הוא מינימלי.

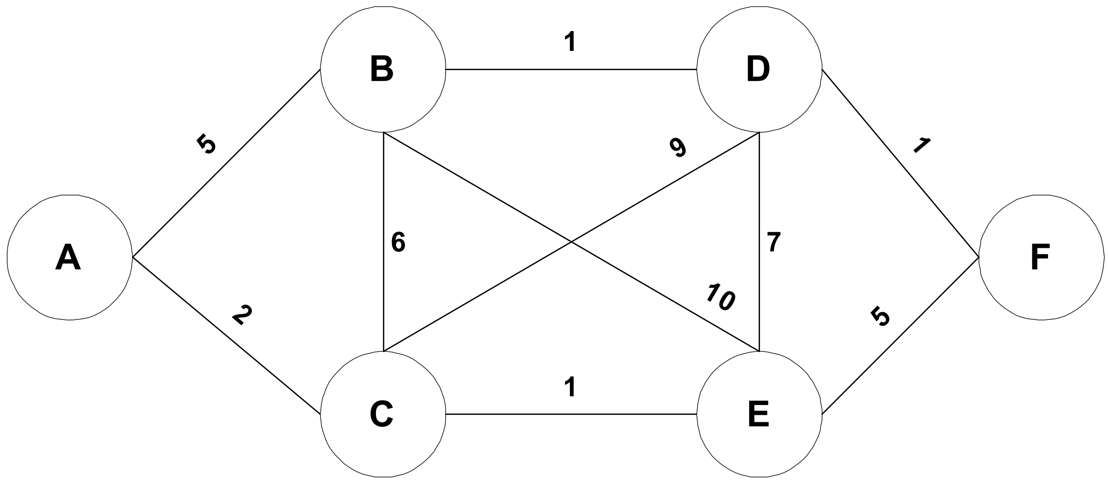
מכוון ש E_T מחברת את כל קודקודי הגרף ולא מכילה מעגלים נראה בהמשך כי T יוצר עץ. עץ כזה נקרא <u>עץ פורש</u>(Spanning Tree).

מטרתנו למצוא עץ פורש T כזה כך שסכום המשקולות המיוחסות לקשתות העץ T, $w(E_T)$,יהיה מינימלי.מציאת עץ T כזה מכונה בשם <u>בעיית העץ הפורש המינימלי</u>.

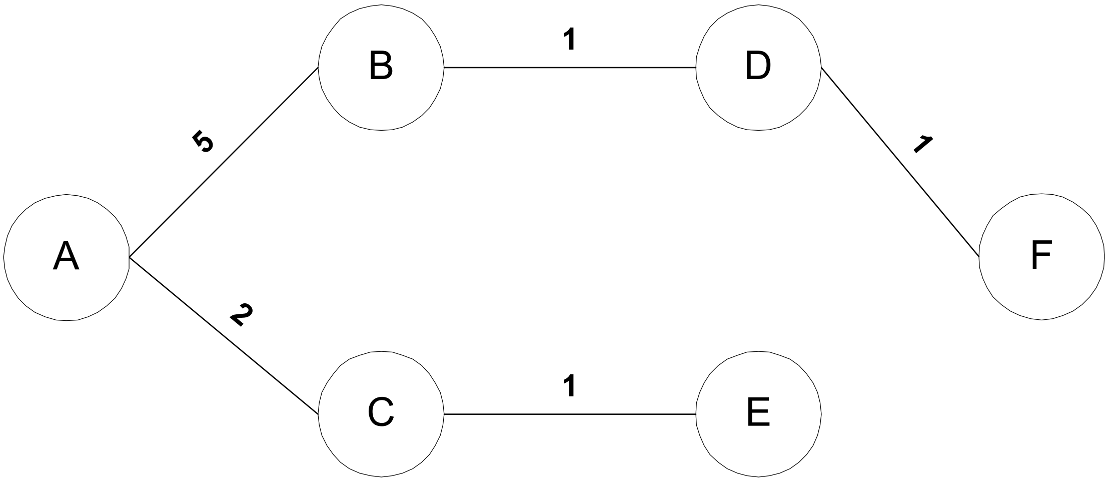
הגדרה:
עץ פורש (Spanning Tree)של גרף G קשיר הוא עץ אשר מכיל את כל קודקודי הגרף G , ועץ זה הוא גם כן גרף קשיר.

הגדרה:
עץ פורש מינימלי (Minimum Spanning Tree)-הינו עץ פורש וסכום המשקולות הרשומות בצד הקשתות של העץ הינו מינימלי.

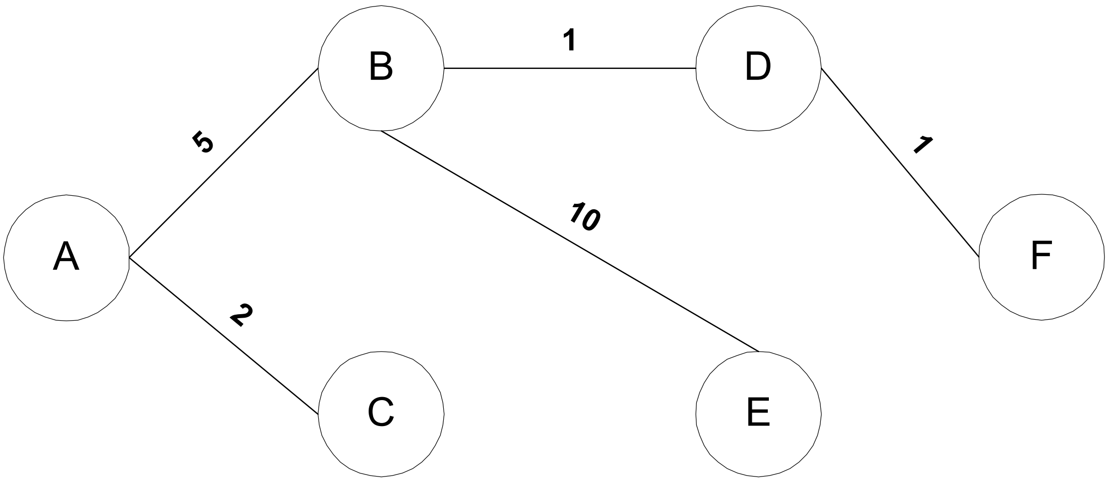
נבהיר את הבעיה בעזרת התרשים הבא :
נתונה ברשת הבאה:



הקודקודים שברשת (A-F) מייצגים את כל היישובים.
כאמור w(u,v) מייצג משקל שעל הקשת (u,v). נניח כי w(u,v) מייצג את המימון הכספי לסלילת כביש בין הישובים u ו v. כאמור הבעיה למצוא עץ בעלות הוצאות מינימליות , כך שכל יישוב חייב להיות מחובר למערכת תחבורתית .
אם נתבונן בתרשימים הבאים :



(a)



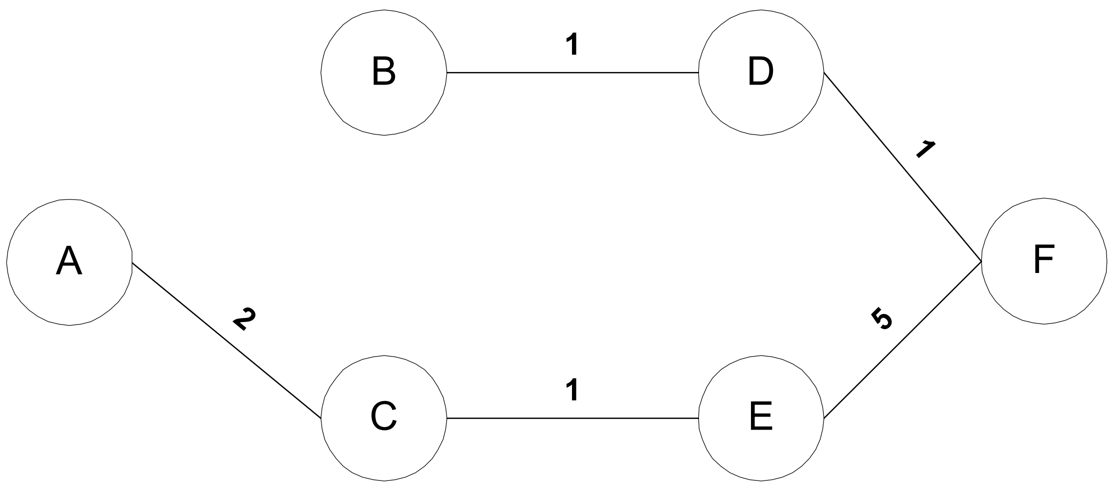
(b)

ברור כי שני התרשימים הם עצים פורשים של הגרף הנתון.
תרשים (a) שהינו עץ יותר טוב מתרשים (b) שהינו גם כן עץ,
מאחר שבתרשים (a) סך ההוצאות שווה ל- 10 (=1+1+1+5+2),
בעוד שבתרשים (b) סך ההוצאות שווה ל- 19 (=1+10+1+2+5).

קל לבדוק שלא ניתן למצוא עץ טוב יותר מזו שבתרשים (a) מבחינת סך ההוצאות למימון סלילת כבישים בין היישובים, כך שכל יישוב חייב להיות מחובר למערכת הכבישים.פתרון כזה מכונה בשם <u>פתרון אופטימלי</u> . בעיית העץ הפורש המינימלי שייכת לבעיות אופימיזציה

(optimization problem),שיש להן פתרונות אפשריים רבים (התרשימים (a) ו (b) הם פתרונות אפשריים כי בשני התרשימים כל היישובים מחוברים למערכת הכבישים). לכל פתרון יש ערך (בבעיה שלנו הערך היינו סך ההוצאות לסלילת מערכת הכבישים), ואנו רוצים למצוא את הפתרון בעל הערך האופטימלי-המקסימלי או המינימלי (בבעיה שלנו סך ההוצאות לסלילת מערכת הכבישים צריך להיות מינימלי).

פתרון אופטימלי אינו <u>ה</u>פתרון <u>ה</u>אופטימלי, מאחר שייתכן שיהיו מספר פתרונות שערכם אופטימלי. כפי שראינו התרשים (a) הינו פתרון אופטימלי.כמו כן התרשים הבא גם כן פתרון אופטימלי:



"סך ההוצאות" ברשת זו גם כן שווה ל-10(=2+1+1+1+5).
ערך פתרון זה אינו טוב ואינו גרוע מערך הפתרון שבתרשים (a).
קל להשתכנע שאי אפשר לשפר את ערך הפתרון שבתרשימים (a) ו (~~b~~) ופתרון (a) או (c) יקרא פתרון אופטימלי.

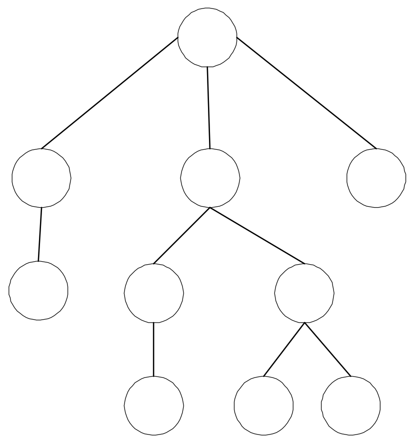
התרשימים (b), (a) ו (c) מייצגים עצים פורשים וקל לראות שניתן למצוא עצים פורשים נוספים עבור הרשת הנתונה.

אך מבין כל העצים הפורשים רק התרשימים (a) ו (c) הם עצים פורשים מינימליים (בדוק!).
בפרק זה נבחן שני אלגוריתמים הפותרים את <u>בעיית העץ הפורש המינימלי</u>:אלגוריתם של קרוסקל (Kruskal) והאלגוריתם של פרים (Prim).
לפני שנכיר את האלגוריתם של קרוסקל ופרים נראה את תכונות יסוד של עצים.

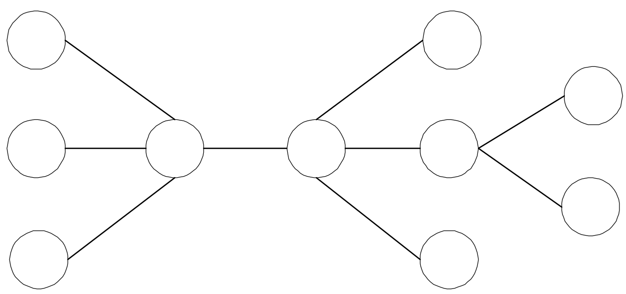
6.2 הגדרות ותכונות יסוד של עצים

<u>הגדרה</u>: גרף G יקרא עץ אם הוא גרף קשיר וחסר מעגלים.

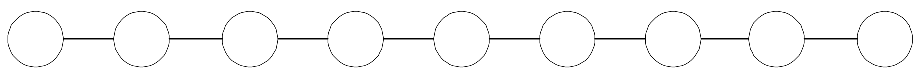
דוגמאות לגרפים שהם עצים :
א.



ב.

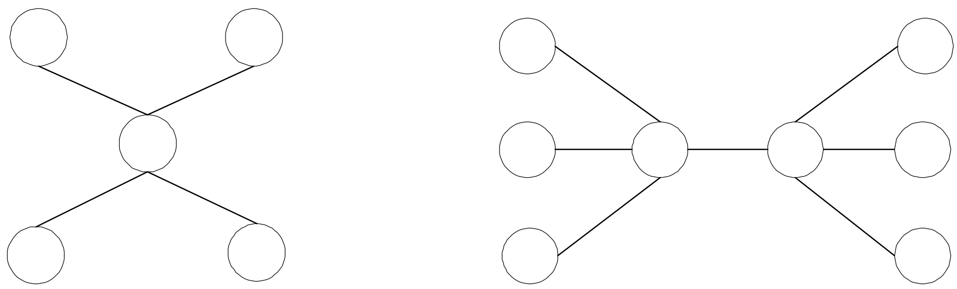


ג.



הגדרה: גרף G יקרא יער אם הוא גרף לא קשיר וכל רכיב בו הוא עץ.

דוגמא:



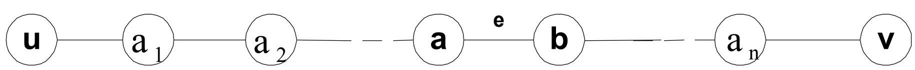
בגרף הלא קשיר הנתון שני רכיבים וכל רכיב הוא עץ.

משפט 6.2.1

בכל עץ T , בין כל שני קודקודים כלשהם u,v קיים מסלול אחד ויחיד.

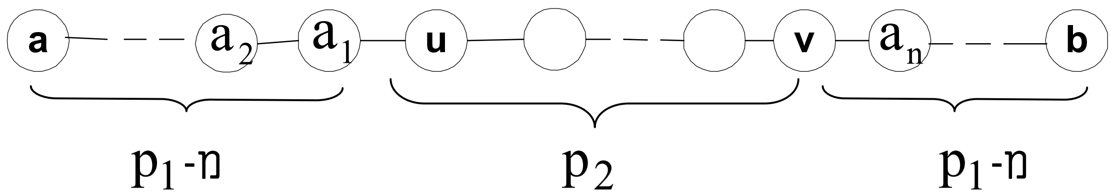
הוכחה:
לפי ההגדרה כל עץ T הוא גרף קשיר , כלומר בין כל שני קודקודים ב-T קיים מסלול ובפרט בין u ל-v קיים מסלול אותו נסמן ב- P. עתה נראה שמסלול זה הוא מסלול יחיד.
נראה את זה בדרך השלילה, כלומר נניח כי בין u ל-v יש לפחות שני מסלולים שונים.כתוצאה מההנחה נגיע לסתירה ולכן המסקנה תהיה בין שני קודקודים כלשהם u ו v לא קיים יותר ממסלול אחד, ולכן הוא יחיד.
הבא נניח כי בין u ל- v קיימים שני מסלולים שונים ונסמנם ב- P_1 ו P_2 .

P_1 :

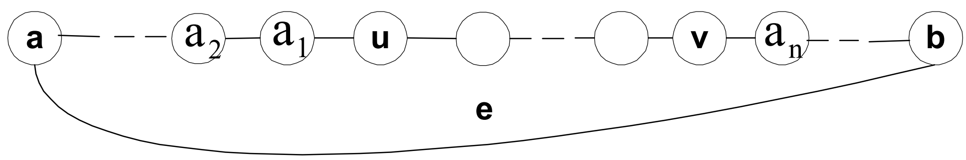


P_2

מסלול P_1 עובר דרך הקשת $e = (a,b)$ ואילו המסלול P_2 אינו עובר דרך הקשת $e = (a,b)$.
נסיר את הקשת $e = (a,b)$ מ-T בפרט מ-P_1. למרות זאת קל לראות שעדין קיים מסלול בין a ל- b והוא המסלול הבא :



זהו מסלול אפשרי בין a ל- b בעץ T . מסלול זה מתאר תת גרף של הגרף הנתון שהוא עץ $T = (V, E_T)$. עתה אם נוסיף את הקשת $e = (a,b)$ במקום שהיה במקור בעץ T אז נקבל תת גרף:



מחד קל לראות שגרף זה שנוצר יש בו מעגל.ומאידך גרף זה שנוצר הוא תת גרף של הגרף הנתון שהוא עץ T. לא יתכן שתת גרף שהינו עץ יכיל מעגל מיכוון שבעץ אין מעגלים.
זוהי סתירה להנחה שבין שני קודקודים כלשהם בעץ T קיים יותר ממסלול אחד. לכן, מספר המסלולים בין שני קודקודי העץ T הוא אחד ויחיד.

משפט 6.2.2

נתון עץ פורש $T=(V,E_T)$. הגרף $G^{'}=(V,E^{'})$ שבו $E^{'}=E_T-e$, כאשר $e=(a,b)$ קשת כלשהי כלומר $e=(a,b)\in E_T$, הוא יער המכיל בדיוק שני עצים.

הוכחה:

קשת $e=(a,b)$ הוא מסלול המחבר בין a ו- b. לפי משפט 6.2.1 מסלול זה הוא אחד ויחיד.

לכן בגרף $G^{'}=(V,E^{'})$, כפי שהוא מוגדר , אין אף מסלול המחבר בין a ל- b ולכן הגרף החדש $G^{'}$ הוא גרף לא קשיר. מאחר שרק a ו- b נותקו אז ב-$G^{'}$ יש בדיוק 2 רכיבים.מאחר ש- T הוא עץ והקשת בין a ל- b נותקה אזי נקבל 2 רכיבים וכל רכיב הוא עץ .לכן $G^{'}$ הוא יער. מש"ל.

משפט 6.2.3

נתון גרף $G=(V,E)$. עבור עץ פורש $T=(V,E_T)$ של הגרף G מתקיים : $|E_T|=|V|-1$,כלומר מספר הקשתות בעץ T קטן באחד ממספר קודקודיו.

הוכחה:

נוכיח את המשפט תוך שימוש באינדוקציה מתימטית על מספר הקודקודים בעץ.

נסמן ב- T_n עץ בעל $|V|=n$ קודקודים.

צריך להוכיח כי מספר הקשתות ב- T_n והוא n-1 .

צעד בסיסי:

עבור $n=1$, ב- T_1, שהוא העץ הבא : O, יש רק קודקוד אחד ואפס קשתות ולכן הטענה מתקיימת.

עבור $n = 2$, ב- T_2 , שהוא העץ הבא O–O, יש בו שני קודקודים וקשת אחת ולכן הטענה מתקיימת.

צעד משלים:

הבא נניח את נכונות הטענה עבור כל K שקטן מ- n כלומר מספר הקשתות ב- T_K הוא $K-1$ לכל $K < n$.

ונוכיח מספר הקשתות ב- T_n הוא n-1 .

עתה יהי $T_n = (V, E_T)$ המכיל n קודקודים. בעץ T_n נבחר קשת כלשהי $e = (a,b)$.

לפי משפט 6.2.2 הגרף $G' = (V, E_T - e)$ הוא יער המכיל 2 רכיבים וכל רכיב הוא עץ. קשת $e = (a,b)$ הוא מסלול המחבר בין a ל- b.לפי משפט 6.2.1 מסלול זה הוא אחד ויחד.מאחר שהקשת $e = (a,b)$ נותקה והיא לא קיימת ביער G' אזי אין אף מסלול המחבר בין a ו- b ביער G'. לכן לאור האמור לעיל a נמצא בעץ אחד ו- b נמצא בעץ השני.

נניח שבעץ אחד כזה יש k_1 קודקודים והוא יסומן כ - T_{k_1} ובשני יש k_2 קודקודים והוא יסומן כ- T_{k_2}.

ברור כי $k_1 + k_2 = n$ ו $k_2 < n$ ו $k_1 < n$.

לכן לפי הנחת האינדוקציה נקבל :

מספר הקשתות ב- T_{k_1} הוא $k_1 - 1$

ומספר הקשתות ב- T_{k_2} הוא $k_2 - 1$.

סופית מספר הקשתות ב- T_n הוא חיבור של מספר הקשתות ב- T_{k_1} ושל מספר הקשתות ב- T_{k_2} בתוספת 1.כלומר מספר הקשתות ב- T_n הוא :

$$(k_1 - 1) + (k_2 - 1) + 1 = k_1 + k_2 - 1 = n - 1$$

מש"ל.

<u>משפט 6.2.4</u>

גרף G הוא עץ $T = (V, E_T)$ אם ורק אם G חסר מעגלים והוספת קשת $e = (a,b)$ בין שני קודקודים לא

מחוברים b ו a ב-G יוצרת מעגל אחד ויחיד , כלומר בגרף $G^{'}=(V,E_T\cup(a,b))$ כאשר $(a,b)\notin E_T$, קיים מעגל אחד ויחיד המכיל את הקשת (a,b) .

הוכחה:

<u>כיוון ראשון</u>

נתון : G חסר מעגלים והוספת קשת e = (a,b) יוצרת מעגל אחד ויחיד.

צריך להוכיח ש- G הוא עץ.

מאחר שנתון כי G חסר מעגלים אז מספיק להראות ש-G הוא גרף קשיר ומכאן נסיק כי G הוא עץ. עתה נותר להראות ש-G הוא גרף קשיר. נוכיח את זה בשלילה.

נניח ש- G הוא גרף לא קשיר ונניח שיש בו שני רכיבים G_2 ו G_1 .

יהיו שני קודקודים כלשהם a,b בגרף הנתון כך ש- $b\in G_2$ -ו $a\in G_1$. לפי הנתונים ברור כי הקשת (a,b) לא קיימת בגרף G ולא קיים מסלול בין a ל-b בגרף G.

עתה אם נוסיף את הקשת e = (a,b) לאוסף הקשתות שבגרף $G=(V,E_T)$ נקבל גרף חדש $G'=(V,E_T\cup(a,b))$. בגרף G' אין מעגל.

אחרת אם בגרף G' יהיה מעגל אז פרושו שבגרף G קיים מסלול בין a ל-b והוספת הקשת החדשה יוצרת מעגל.זה בניגוד להנחה שלא קיים מסלול בין a ל- b הגענו לסתירה ולכן G גרף קשיר.

מסקנה:G היא עץ.

<u>כיוון שני</u>

נתון G עץ.

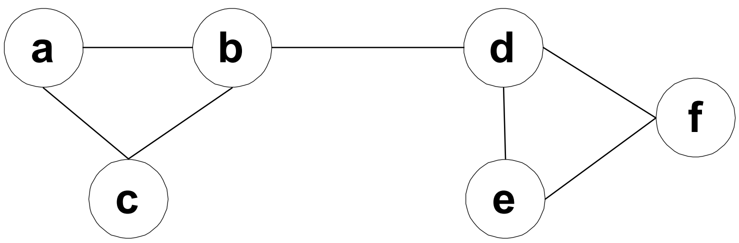
צריך להוכיח G חסר מעגלים והוספת קשת כלשהי = e (a,b) תיצור מעגל אחד ויחיד ב- G .מאחר ש- G היא עץ אז לפי ההגדרה היא גם חסר מעגלים.

בהינתן שני קודקודים כלשהם a,b בגרף (שהוא עץ) קיים מסלול אחד ויחיד בניהם.לכן הוספת קשת e = (a,b) תיצור מעגל יחיד. מכיוון שאם יש יותר ממעגל אחד המכילים את הקשת e = (a,b) אז קיים יותר ממסלול אחד בין a ל-b בעץ הנתון בניגוד למשפט 6.2.1 מש"ל.

הגדרה : נתון גרף בלתי מכוון קשיר G = (V,E) . קשת e בגרף G נקראת <u>קשת מפרידה</u> אם הסרתה בגרף G תיצור גרף שבו שני רכיבים.

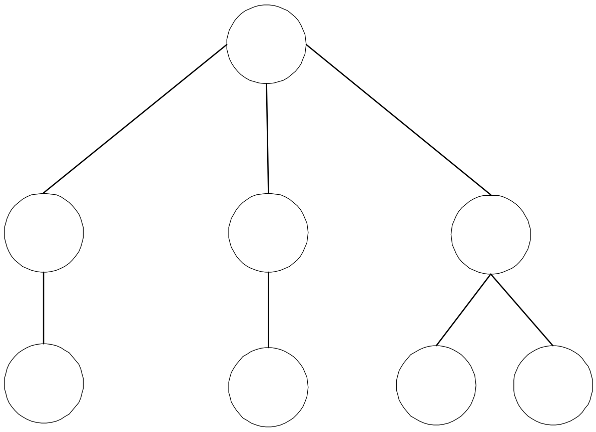
דוגמאות:

1. בגרף הבא :



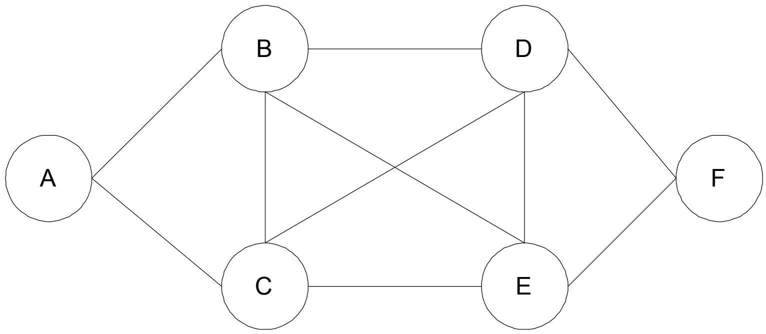
קשת (b,d) הינה קשת מפרידה ויתר הקשתות שבגרף זה אינן מפרידות.

2. בגרף הבא שהינו עץ :



כל קשת היא קשת מפרידה.

3. בגרף הבא :



אין קשת מפרידה.

<u>משפט 6.2.5</u>

נתון גרף בלתי מכוון קשיר $G=(V,E)$.
קשת כלשהי $e \in E$ היא קשת מפרידה אם ורק אם e אינה שייכת לקבוצת הקשתות כלשהי המהוות מעגל בגרף G .
ההוכחה היא פשוטה יחסית ונשארת כתרגיל לקורא.

6.3 <u>אלגוריתם של קרוסקל(kruskal) למציאת עץ פורש מינימלי</u>

באלגוריתם זה בוחרים תחילה את הקשת בעלת משקל מינימלי , לאחר מכן את הקשת בעלת משקל מינימלי מבין הקשתות הנותרות וכן הלאה , בלבד שלא ניצור מעגל.

6.3.1 <u>אלגוריתם של קרוסקל</u>

<u>צעד 1</u>

1.1 $E_T = \emptyset$

(*בתחילת האלגוריתם קבוצת הקשתות בעץ פורש מינימלי הינה קבוצה ריקה*)

1.2 $k \leftarrow 0$

(*עד כה ביצענו אפס צעדים באלגוריתם *)

<u>צעד 2</u>

2.1 מצא קשת a השייכת ל-$(E-E_T)$ כך שלכל קשת b אשר שייכת ל -$(E-E_T)$ מתקיים :

$$w(a) = \min_{b} w(b)$$

(*כלומר מאתרים קשת , מקבוצת E בגרף G ,
שהינה בעלת משקל מינימלי ולא נבחרה עד כה*)

2.2 אם לא קיימת קשת כזו אזי אלגוריתם מסתיים ! .
אחרת

2.2.1 אם קבוצה $E_T \cup \{a\}$ יוצרת מעגל

2.2.1.1 אז בצע :

2.2.1.1.1 $E \leftarrow E - \{a\}$

2.2.1.1.2 חזור לצעד 2.1

2.2.1.2 אחרת בצע :

2.2.1.2.1 $E \leftarrow E - \{a\}$

2.2.1.2.2 $E_T \leftarrow E_T + \{a\}$

2.3 אם $k = |V|\text{-}1$ אזי הגרף $G = (V, E_T)$ הינו עץ פורש מינימלי ועוצר את האלגוריתם.

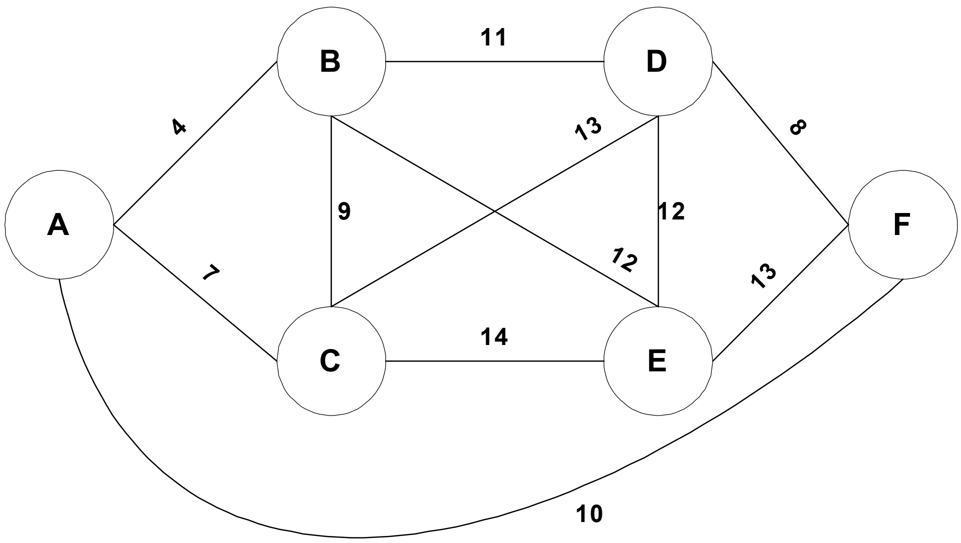
2.4 אם $k < |V|\text{-}1$ והאלגוריתם הסתיים אזי עצור את האלגוריתם ותודיע : "לא קיים עץ פורש ב-G ".

<u>צעד 3</u>

3.1 $k \leftarrow k + 1$

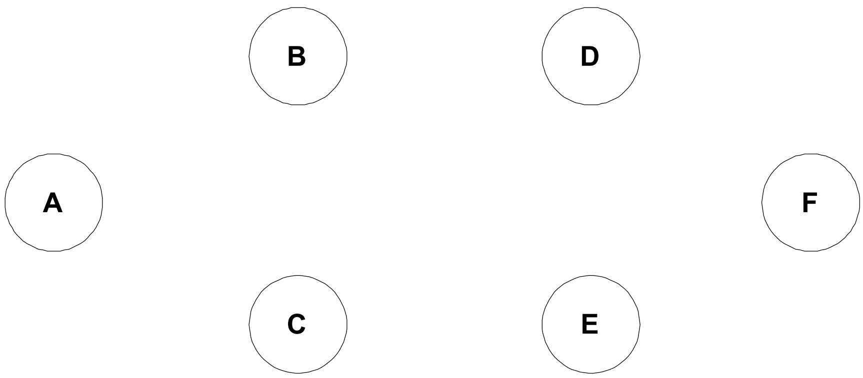
3.2 חזור לצעד 2

נדגים את פעולת האלגוריתם של קרוסקל באמצעות הגרף שמופיע בתרשים הבא:

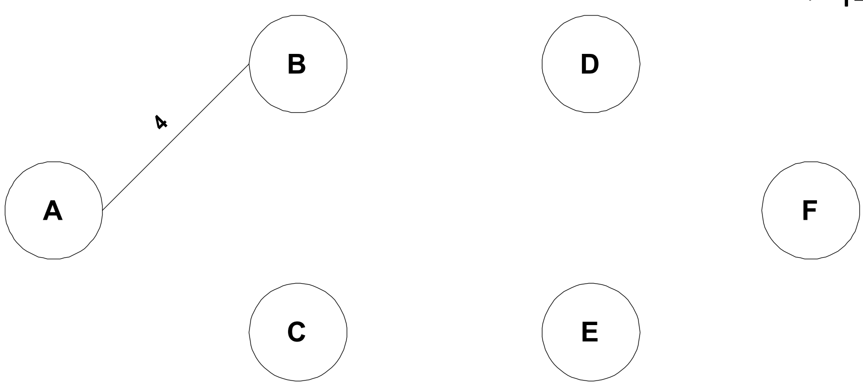


בגרף זה 6 קודקודים , לכן בעץ הפורש יהיו 5 קשתות (לפי משפט 6.2.3) .

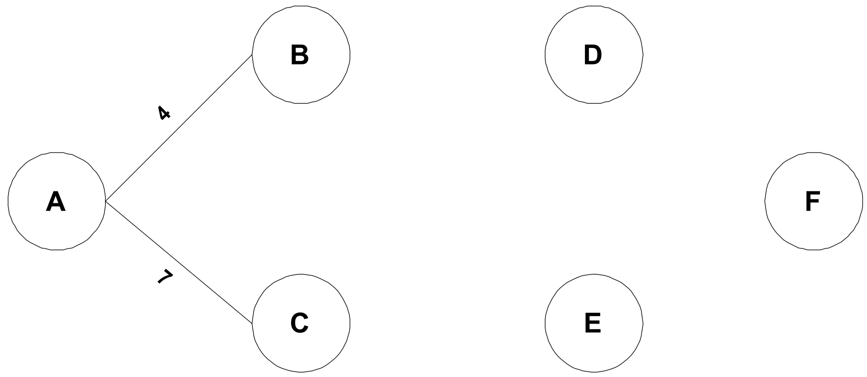
בהתחלה מתחילים עם יער שבו כל הקודקודים אינם מחוברים בניהם , לכן נקבל :



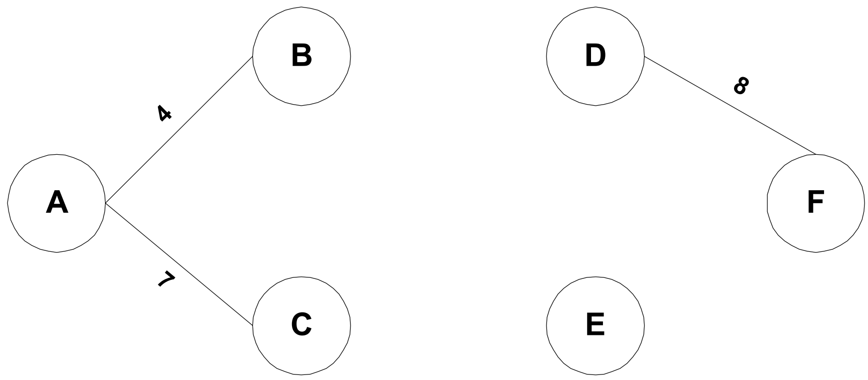
באיטרציה ראשונה מוצאים את הקשת (A,B) שהינה בעלת המשקל הקטן ביותר.לכן בתום האיטרציה הראשונה היער יראה כך :



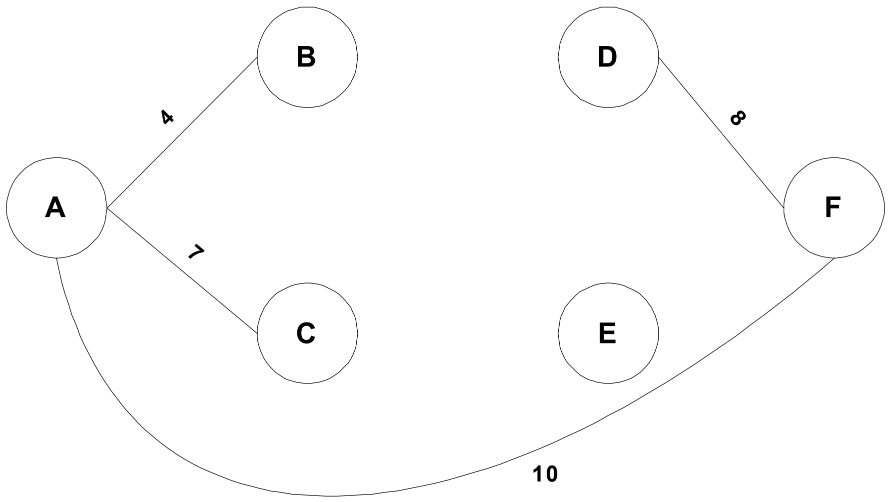
באיטרציה שניה מוצאים את הקשת (A,C) שהינה בעלת המשקל הקטן ביותר מבין הקשתות שאינן שייכות ליער.
לכן בתום האיטרציה השניה היער יראה כך :



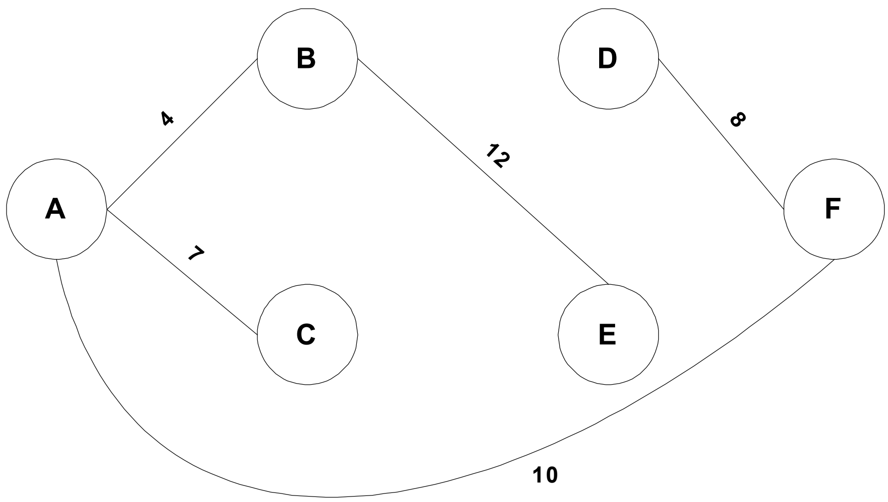
באיטרציה השלישית מוצאים את הקשת (D,F) שהינה בעלת המשקל הקטן ביותר מבין הקשתות שאינן שייכות ליער.
לכן בתום האיטרציה השלישית היער יראה כך :



באיטרציה רביעית מוצאים את הקשת (B,C) שהינה בעלת המשקל הקטן ביותר מבין הקשתות שאינן שייכות ליער.
אך לא נוסיף ליער את הקשת (B,C) , מכיוון שהוספתה יצור מעגל.לכן עתה נמצא את הקשת (A,F) שהינה בעלת המשקל הקטן ביותר מבין הקשתות שאינן שייכות ליער.לכן בתום האיטרציה הרביעית היער יראה כך :



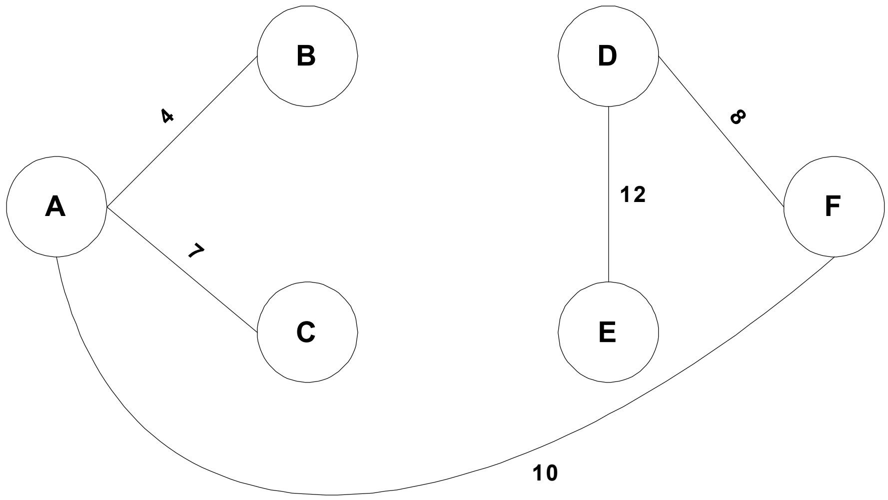
באיטרציה חמישית מוצאים את הקשת (B,D) שהינה בעלת המשקל הקטן ביותר מבין הקשתות שאינן שייכות ליער.
אך לא נוסיף ליער את הקשת (B,D) , מיכוון שהוספתה יצור מעגל.לכן עתה נמצא את הקשת (B,E) שהינה בעלת המשקל הקטן ביותר מבין הקשתות שאינן שייכות ליער.לכן בתום האיטרציה החמישית היער יראה כך :



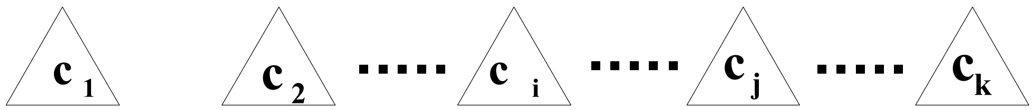
כמובטח לאחר חמש איטרציות קיבלנו עץ פורש מינימלי.

שים לב לכך , שעץ פורש מינימלי אינו בהכרח אחד ויחיד.
לדוגמא , באיטרציה החמישית אפשרי לבחור את הקשת

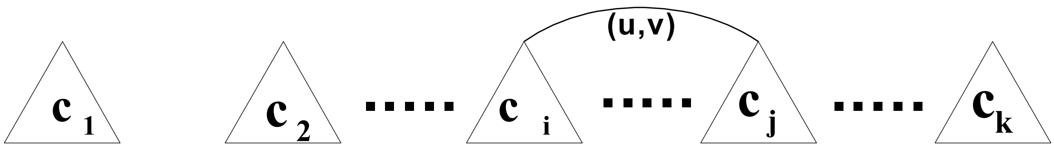
(D,E) , שהינה בעלת משקל 12 ולא את הקשת (B,E) שנבחרה וגם לה מיוחס משקל 12. לכן בתרשים הבא מוצג עץ פורש מינימלי אחר בשביל אותו גרף .



כאמור בכל צעד אלגוריתם של קרוסקל מוסיף ליער קשת בעלת משקל קטן ככל האפשר בתנאי שהוספתה לא תיצור מעגל.עתה השאלה העומדת בפנינו:
כיצד נוכל לקבוע האם הקשת הנבחנת שהינה בעלת משקל קטן ככל האפשר הוספתה לעץ פורש לא תיצור מעגל ?
כזכור בהתחלה אלגוריתם של קרוסקל יוצר $|V|$ רכיבי קשירות (יער) עבור הגרף $G = (V,E)$.
בכל שלב של האלגוריתם מוסיפים ליער קשת (A,B) שהינה בעלת משקל מינימלי מבין כל הקשתות המחברות שני עצים כלשהם ביער.אם לאחר מספר איטרציות ביער ישנם k עצים זרים זה לזה :



אזי לאחר הוספת הקשת (u,v) ביער יהיו k-1 עצים זרים זה לזה.



שים לב לכך , שאם $u \in c_i$ ו $v \in c_j$ כאשר $i \neq j$ אזי הוספת הקשת (u,v) לא תיצור מעגל , אך אם $u \in c_i$ וגם $v \in c_i$ בעבור i כלשהו , אזי הוספת הקשת (u,v) תיצור מעגל לפי משפט 6.2.4.

בכל שלב של אלגוריתם קרוסקל , פרט לשלב האחרון , אנו מקבלים יער שהוא גרף לא קשיר וכל רכיב בו הוא עץ.
כל רכיב הינו קבוצה של קודקודים והרכיבים הם זרים זה לזה.לכן יש צורך לייצג את היער באמצעות מבנה נתונים המתחזק קבוצות זרות של איברים.כל קבוצה תכיל את קודקודיו של עץ אחד ביער.
על מבנה זה נגדיר את מספר הפעולות הבסיסיות הבאות:

MAKE-SET(v) – יצירת רכיב קשירות חזקה כקבוצה בת איבר אחד – $\{v\}$.

FIND(v) – פעולה זו מחזירה מספר הקבוצה , אליה שייך הקודקוד v .

UNION(u,v) – פעולה זו מקבלת 2 רכיבים קשירים u ו v וגורמת לקריסת 2 רכיבים אלו לרכיב קשירות אחת .

SORT(E) – פעולה זו ממיינת את הקשתות לפי סדר לא יורד על סמך המשקולות שעליהן.

לאור האמור לעיל , לפניך אלגוריתם מילולי למימוש האלגוריתם של קרוסקל ולאחר מכן מימוש האלגוריתם של קרוסקל תוך שימוש בפעולות הבסיסיות המוגדרות על טיפוס נתון יער.

<u>אלגוריתם מילולי</u>

<u>צעד 1</u> בעץ פורש קבוצת הקשתות הינה קבוצה ריקה
$(E_T \leftarrow \emptyset)$

<u>צעד 2</u> צור $|v|$ רכיבי קשירות (עצים), מאחר שכל קודקוד בגרף הנתון מגדיר עץ שהינו קבוצה בת איבר אחד.

<u>צעד 3</u> מיין את קשתות העץ לפי סדר לא יורד על סמך המשקולות המיוחסות להן.

<u>צעד 4</u> לכל קשת (u,v) , לפי הסדר שנקבע בצעד 3 , בצע :

4.1 אם קבוצה שאליה שייך קודקוד u שונה מהקבוצה אליה שייך קודקוד v אז בצע :

3.1.1 $E_T \leftarrow E_T \cup \{(u,v)\}$

3.1.2 שני רכיבי קשירות של u,v קורסים לרכיב קשירות אחת. (UNION(u,v)).

<u>צעד 5</u> להחזיר עץ פורש שהינו E_T.

סופית להלן אלגוריתם של קרוסקל

נתון גרף $G=(V,E)$.
המטרה לבנות עץ פורש מינימלי $T=(V,E_T)$.

KRUSKAL(G)
//PRE-ALGORITHM

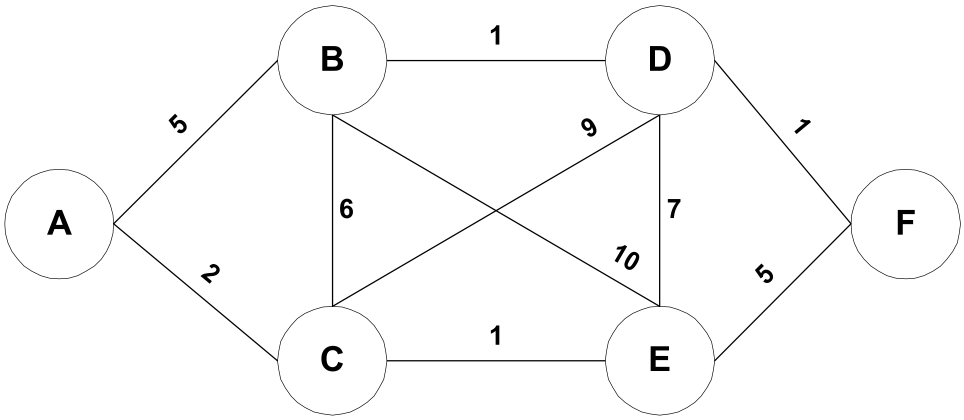
$L \leftarrow \text{SORT}(E)$
for each $v \in V$ do $\text{MAKE-SET}(v)$
$E_T \leftarrow \Phi$

//GROW TREE

for each $(u,v) \in L$ do
    $u' \leftarrow \text{FIND}(u)$
    $v' \leftarrow \text{FIND}(v)$
    if $u' \neq v'$ then
        $E_T \leftarrow E_T \cup \{(u,v)\}$
        $\text{UNION}(u',v')$
return $G(V,E_T)$

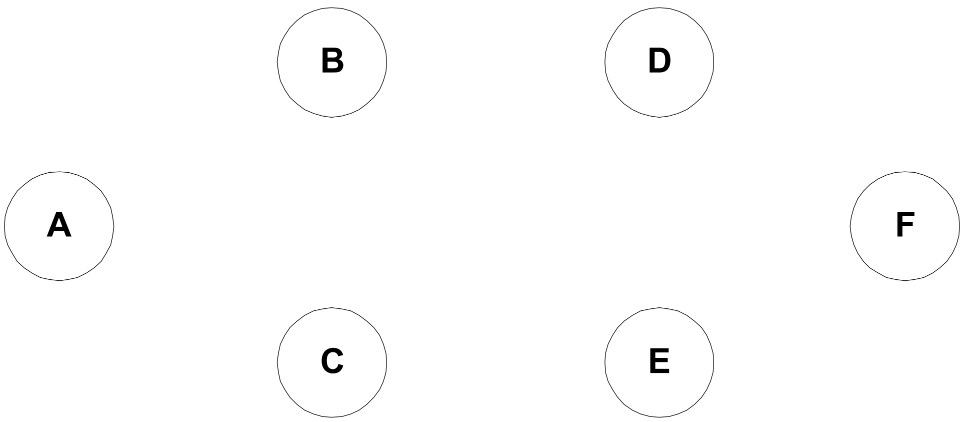
דוגמא למציאת עץ פורש מינימלי בעזרת אלגוריתם Kruskal

נדגים את התהליך שתואר לעיל למציאת עץ פורש מינימלי על הרשת (גרף משוקלל) שבתרשים הבא :



צעד 1+2

עץ פורש T הינה קבוצה ריקה של קשתות.ושישה רכיבי קשירות.תמונת המצב הינה:



<u>צעד 3</u>

כתוצאה מהמיון של קשתות נקבל רשימה L :

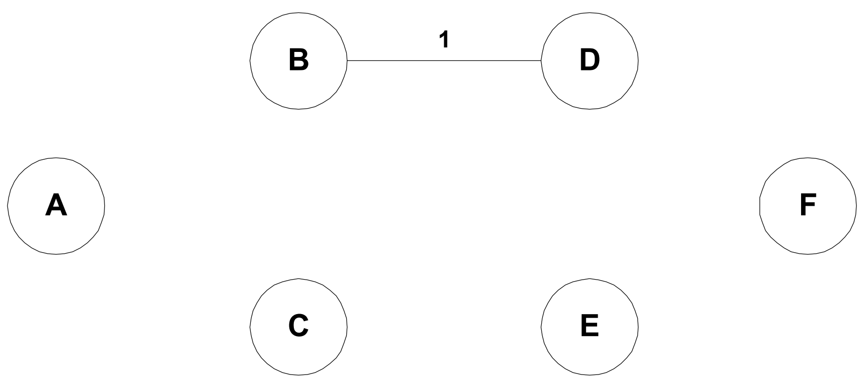
(B,D) 1
(C,E) 1
(D,F) 1
(A,C) 2
(A,B) 5
(E,F) 5
(B,C) 6
(D,E) 7
(C,D) 9
(B,E) 10

<u>צעד 4</u>

סורקים את הרשימה L מתחילתה עד סופה, לפי הסדר שנקבע בצעד 3 כתוצאה מהמיון, ומטפלים בכל הקשתות שב- L .

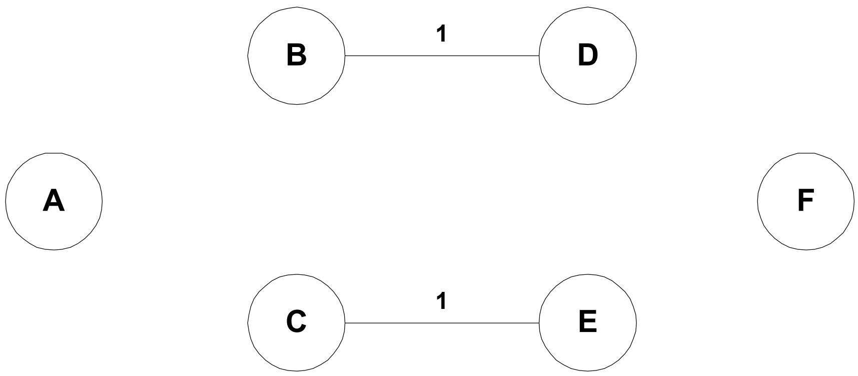
<u>הקשת המטופלת (B,D)</u>

מאחר ש- B ו- D שייכים לשני רכיבי קשירות זרים אז B ו- D יקרסו לרכיב קשירות אחד ונקבל :



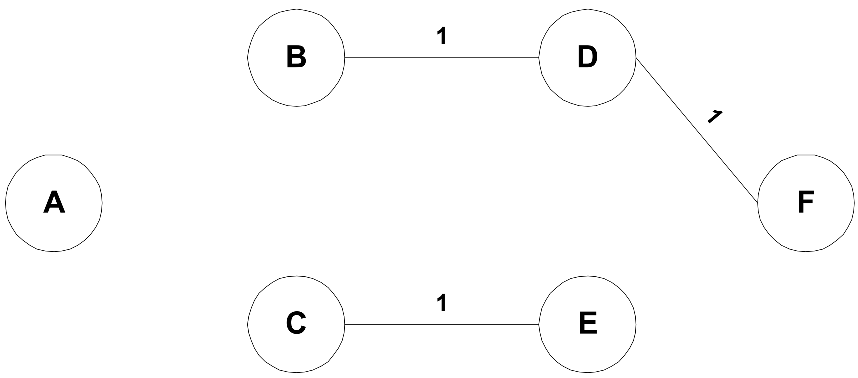
<u>הקשתות המטופלת (C,E)</u>

מאחר ש- C ו- E שייכים לשני רכיבי קשירות זרים אז C ו- E יקרסו לרכיב קשירות אחד ונקבל :

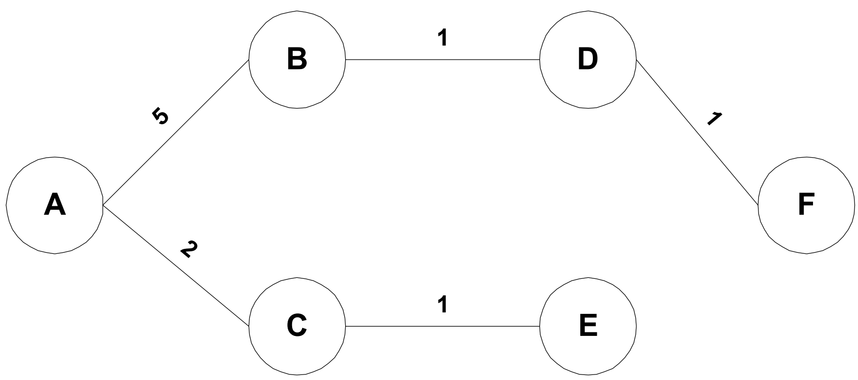


<u>הקשתות המטופלת (D,F)</u>

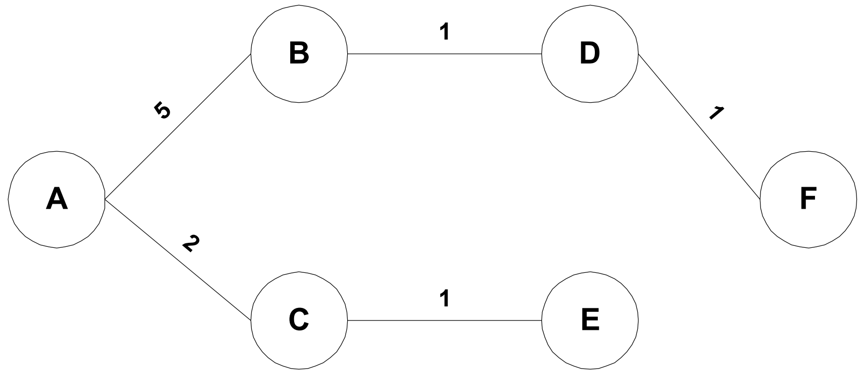
מאחר ש- D ו- F שייכים לשני רכיבי קשירות זרים , הם קורסים לרכיב קשירות אחד ונקבל :



כך ממשיכים ולאחר שתי איטרציות תמונת המצב הינה:



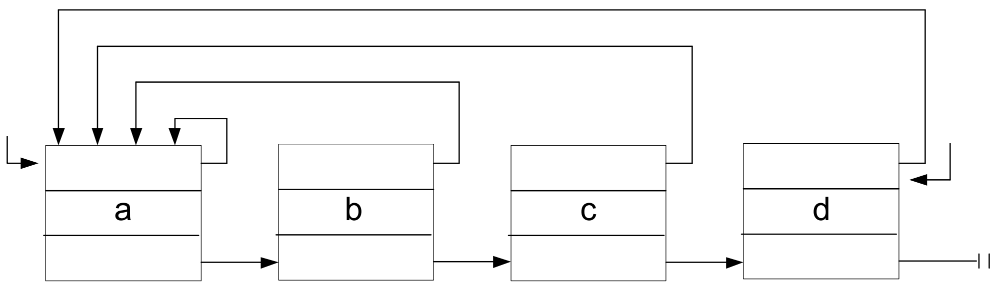
כעת הקשת המטופלת היא (E,F)
מאחר ש- E ו- F בשלב זה שייכים לאותו רכיב קשיר לא נוסיף את הקשת (E,F) לעץ פורש T. באופן אנלוגי הקשתות הבאות , שברשימה L , (B,C) , (D,E) , (C,D) , (B,E) לא תתווספנה לעץ פורש T וסופית עץ פורש מינימלי שמתקבל כצפוי הינו :

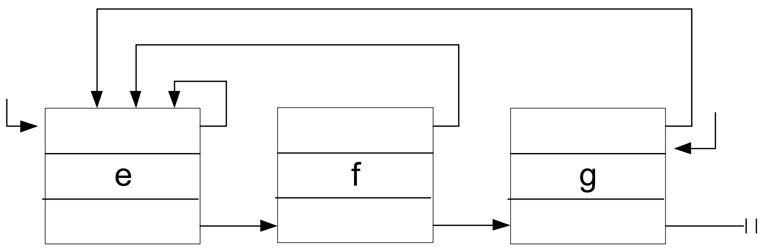


6.3.2 יעילות האלגוריתם של קרוסקל

קל לראות כי זמן הריצה של האלגוריתם – קרוסקל , אשר פועל על גרף בלתי מכוון קשיר $G = (V, E)$, תלוי במימוש מבנה הנתונים יעד שהינו קבוצות זרות.

א. ייצוג קבוצות זרות על ידי רשימות מקושרות.
כל קבוצה מיוצגת על ידי רשימה מקושרת.והאיבר הראשון בכל רשימה משמש כנציג הקבוצה ,המיוצגת באמצעות הרשימה.
התרשים הבא מתאר את הייצוג של שתי קבוצות זרות על ידי שתי רשימות מקושרות:





ברשימה מקושרת כזו כל צומת של הרשימה שייך לאיבר אחד של קבוצה. בכל צומת ברשימה 3 שדות עיקריים:

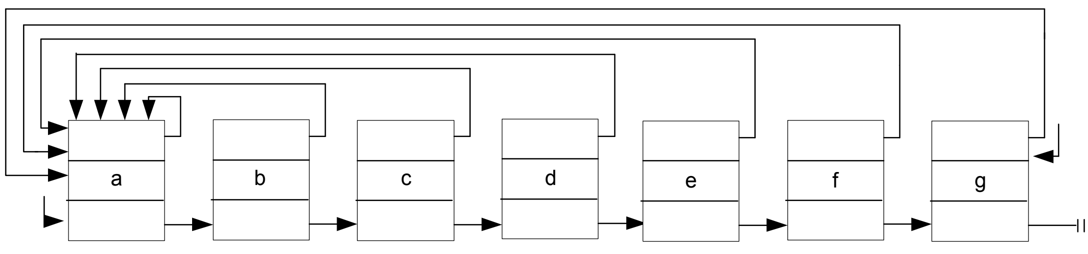
האחד – מצביע לצומת הבא המכיל איבר כלשהו של אותה קבוצה.

השני – מצביע לצומת הראשון ברשימה המכיל איבר שהוא הנציג של הקבוצה.

השלישי – מכיל את האיבר.

שים לב לכך:

1. הרשימה הראשונה מכילה את איברי הקבוצה {a,b,c,d} , והרשימה השניה מכילה את איברי הקבוצה {e,f,g} .
2. לכל רשימה יש מצביע לראשה וגם לסופה.
3. הרשימה המתקבלת מפעולת Union (b,g) היא :



בייצוג כזה של קבוצות זרות על ידי רשימות מקושרות נקבע את סיבוכיות זמן הריצה של הפעולות הבסיסיות המוגדרות על טיפוס נתון יער.

<u>עבור הפעולה</u>	סיבוכיות זמן הריצה <u>במקרה הגרוע ביותר</u>
MAKE_SET(v)	$O(1)$
FIND(v)	$O(1)$
UNION(x,y)	$O(y) \cong O(n^2)$ בדוק!

לאור האמור לעיל עתה נוכל לחקור את סיבוכיות זמן הריצה של האלגוריתם קרוסקל.

<u>צעד</u>	<u>דורש זמן</u>
1.	$O(1)$
2.	$O(\|V\|)$
3.	$O(\|E\|\log\|E\|)$
4.	$O\left(\|E\|\cdot\left(1+\|V\|^2\right)\right)=O\left(\|V\|^2\cdot\|E\|\right)$

לכן סיבוכיות זמן הריצה של האלגוריתם קרוסקל הוא $O\left(|V|^2\cdot|E|\right)$.

במדעי המחשב ברמה אקדמית ניתן לייצג קבוצות זרות באמצעות מבנה נתונים מיוחד הנקרא עצים מושרשים. כאן לא נקיים דיון אודות עצים מושרשים מכיוון שנושא זה חורג מדרישות הקורס . נספר רק על העובדה כי באמצעות מבנה נתונים ״ חכם ״ ניתן לבצע את צעד 4 של האלגוריתם קרוסקל בזמן $O(|E|\log|E|)$ ולכן סיבוכיות זמן הריצה של האלגוריתם הוא: $O(|E|\log|E|)$.

6.3.3 אלגוריתם חמדני (greedy) .

אלגוריתמים הפותרים בעיות ,המכונות כבעיות אופטימיזציה, מבצעים סדרה סופית של צעדים ובכל צעד מקבלים מספר הכרעות בין אפשרויות שונות.
אלגוריתם חמדן בוחר באפשרות הטובה ביותר באותו רגע בתקווה שבחירה כזו תוביל לפתרון אופטימלי כללי. באופן כללי אלגוריתמים חמדניים לא נותנים בהכרח פתרונות אופטימלים כוללים לבעיות, אולם עבור בעיות לא מעטות הם נותנים פתרון אופטימלי כולל , ובפרט גם במקרה של בעיית העץ הפורש המינימלי ניתן להראות שאלגוריתמים חמדניים מסויימים מחזירים עץ פורש בעל משקל מינימלי.
אלגוריתם של קרוסקל מוצא ,בכל שלב של האלגוריתם , קשת (a,b) בעלת משקל מינימלי מבין כל הקשתות המחברות שני עצים כלשהם ביער.מכאן נובע, שהאלגוריתם של קרוסקל הוא אלגוריתם חמדן מכיוון שבכל צעד הוא מוסיף ליער קשת בעלת משקל קטן ככל האפשר. בחירת הקשת בעלת משקל קטן ככל האפשר זוהי אפשרות הטובה ביותר באותו רגע (בכל צעד וצעד) , מכיוון שברצונינו למצוא עץ המחבר בין כל קודקודי הגרף וסכום המשקולות הרשומות (המיוחסות) על הקשתות העץ הוא מינימלי. השאלה המרכזית היא האם גישה חמדנית כזו נותנת פתרון אופטימלי כולל לבעיית עץ פורש מינימלי ?
התשובה לשאלה זו הינה חיובית ונראה בהמשך כי אסטרטגיה חמדנית כפי שבאה לידי ביטוי באלגוריתם של קרוסקל , מחזירה עץ פורש בעל משקל מינימלי.

6.3.4 נכונות אלגוריתם של קרוסקל

מחד האלגוריתם של קרוסקל מתבסס על גישה חמדנית שאינה מבטיחה תמיד פתרון אופטימלי כולל, ומאידך, נאמר שאלגוריתם של קרוסקל מניב עץ פורש מינימלי . המשפט הבא מראה את הנכונות של טענה זו.

<u>משפט 6.3.4.1</u>

לאחר הרצת אלגוריתם של קרוסקל על הגרף G=(V,E) מתקיים :

א. מספר הצעדים באלגוריתם של קרוסקל הוא $|V|-1$ לכל היותר .

ב. אם אלגוריתם מסתיים לאחר m איטרציות אזי:

אם $m=|V|-1$ אזי $T=(V,E_T)$ הינו העץ הפורש המינימלי של G .

אחרת $(m<|V|-1)$ גרף G לא מכיל עץ פורש.

<u>הוכחה:</u>

א. לפי משפט 6.2.3 מספר הקשתות בעץ פורש T הינו$|V|-1$, כלומר $|E_T|=|V|-1$.לכן, אם קיים עץ פורש בגרף G אזי מספר הצעדים הנדרשים למציאת עץ פורש הוא $|V|-1$, אחרת האלגוריתם יסתיים לפני כן ומספר הצעדים יהיה קטן מ- $|V|-1$.

ב. מאחר שבגרף הנתון G יש $|V|$ קודקודים אזי לפי ההגדרה גם בעץ הפורש חייב שיהיו $|V|$ קודקודים. לכן , לפי משפט 6.2.3 בעץ פורש T של G תהיינה $|V|-1$ קשתות.

בכל צעד של האלגוריתם של קרוסקל מוסיפים ליער רק קשת אחת . לכן אם האלגוריתם מסתיים לאחר m איטרציות ו- $m<|V|-1$ אזי יוחזר על ידי האלגוריתם יער שמכיל יותר מעץ אחד ולא את העץ הפורש המבוקש. משמעות הדבר שלא קיימת קשת שתחבר בין העצים השונים ביער , משמע הגרף לא קשיר מכיוון שגרף G מכיל עץ פורש אם ורק אם הוא קשיר . כאשר $m=|V|-1$, האם האלגוריתם של קרוסקל מניב עץ פורש מינימלי ? יהי T עץ פורש של G שנבנה באמצעות האלגוריתם של קרוסקל , ונניח שצלעותיו הן :

$$\{e_1,e_2,\ldots,e_{m-1}\}$$

עתה נניח ש- T אינו אופטימלי , כלומר אינו עץ פורש מינימלי.

נבחר עץ אופטימלי (עץ פורש מינימלי) T1 מבין העצים
הפורשים המינימליים של G , כך שהעץ T1 יכיל
המספר הגדול ביותר של קשתות השייכים ל- T באופן
הבא : $\{e_1, e_2, ..., e_{m-1}\}$ שייכות ל- T וגם ל- T1 ו- $\mathbf{e_m}$
הקשת הראשונה שעבורה מתקיים :

(*) $\mathbf{e_m \in T}$ אך $\mathbf{e_m \notin T1}$ ולא קיים עץ פורש מינימלי אחד
של G אשר מכיל את $\mathbf{e_m}$.

נסמן ב: $\mathbf{T = (V, E_T)}$ את העץ הפורש של G שמתקבל
באמצעות האלגוריתם של קרוסקל וב- $\mathbf{T1 = (V, E_{T1})}$
את עץ הפורש המינימלי של הגרף G .
$W(T)$ - הינו "משקל" העץ T ואילו
$W(T1)$ - הינו "משקל" העץ T1 .

מאחר ש- T1 הוא עץ ו- $\mathbf{e_m \notin E_{T1}}$ אזי לפי משפט 6.2.4
$\mathbf{E_{T1} + e_m}$ (הוספת קשת $\mathbf{e_m}$ לעץ T1) יצור מעגל C
אחד ויחיד אשר מכיל את קשת $\mathbf{e_m}$.
ברור שלא כל הקשתות היוצרות מעגל C שייכות ל-T,
מכיוון ש-T הוא עץ , שהוא גרף קשיר חסר מעגלים .
תהי קשת $\mathbf{e'_m}$ במעגל C כך ש- : $\mathbf{e'_m \in E_{T1}}$ ו- $\mathbf{e'_m \notin T\cancel{1}}$
וגם $\mathbf{e'_m \neq e_m}$.

(**) לפי האלגוריתם של קרוסקל ברור כי $\mathbf{W(e_m) \le W(e'_m)}$.
כמו כן בגלל ש- $\mathbf{e'_m}$ הינה קשת במעגל C , אזי היא לא
יכולה להיות קשת מפרידה בגרף $\mathbf{G' = (V, E_{T1} + \{e_m\})}$
כלומר הגרף הבא $T'' = (V, E_{T''})$ שבו
$\mathbf{E_{T''} = E_T + e_m - e'_m}$ הוא גרף קשיר וגם עץ , המכיל
$n = |V|$ קודקודים כלומר $n-1$ קשתות.ולכן הוא גם עץ
פורש של G .
לכן

$$W(E_{T''}) = W(E_{T1}) + W(e_m) - W(e'_m) \le W(E_{T1})$$

כלומר סכום המשקולות של קשתות העץ ''T קטן או

שווה מסכום המשקולות של קשתות העץ T1 . אך מכיוון ש-T1 הוא עץ אופטימלי (עץ פורש מינימלי) מזה נובע ש-''T הוא גם כן עץ פורש אופטימלי(עץ פורש מינימלי).
מחד גיסא: ''T עץ פורש מינימלי ו- $\mathbf{e_m} \in \mathbf{E_{T''}}$ ומאידך גיסא: נאמר ב- (*) כי לא קיים עץ פורש מינימלי של G אשר מכיל את $\mathbf{e_m}$. לכן קיבלנו סתירה להנחתינו ש- T אינו עץ אופטימלי . לכן נסיק כי T הוא עץ אופטימלי (עץ פורש מינימלי) .

<u>מש"ל.</u>

6.3.5 גישה כללית לבניית עץ פורש מינימלי ונכונותה.

אלגוריתם של קרוסקל מתחזק קבוצת קשתות $\mathbf{E_T}$ המוכלות בעץ פורש מינימלי. בכל צעד האלגוריתם מוצא קשת (a,b) כזו כך שהוספתה ל $\mathbf{E_T}$, כלומר $\mathbf{E_T + \{ (a, b) \}}$, הינה עדיין תת קבוצה של קשתות המוכלות בעץ פורש מינימלי. באמצעות אסטרטגיה חמדנית זו בונים עץ פורש מינימלי.

באלגוריתם של קרוסקל $\mathbf{E_T}$ הוא יער והאסטרטגיה היא שבכל שלב מוסיפים ל $\mathbf{E_T}$ קשת בעלת משקל מינימלי אשר מחברת שני רכיבים זרים.

לעומת זאת, באלגוריתם של פרים, אותו נכיר בהמשך $\mathbf{E_T}$ הוא עץ ולא יער והאסטרטגיה היא שבכל שלב מוסיפים ל $\mathbf{E_T}$ קשת בעלת משקל מינימלי אשר מחברת את העץ T לקודקוד שלא שייך לעץ T.

לכן ניתן לנסח את האלגוריתם הכללי למציאת עץ פורש מינימלי כדלכמן:

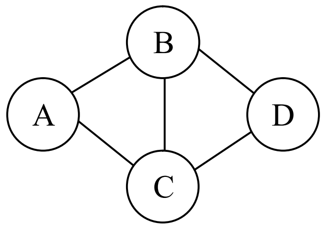
1. $(E_T \leftarrow \emptyset)$
2. בצע לולאה $\mathbf{1 - |V|}$ פעמים על:
   2.1 מצא קשת (a,b) לפי האסטרטגיה המתאימה כך ש $\mathbf{E_T + \{ (a, b) \}}$ הינה תת קבוצה של קשתות של עץ פורש מינימלי.
   2.2 $\mathbf{E_T \leftarrow E_T + \{(a, b)\}}$.
3. החזר את E_T.

הגדרה

חתך בגרף לא מכוון קשיר G=(V,E) מחלק את קבוצת הקודקודים V לשתי תתי קבוצות זרות X ו $\bar{X}$ כך ש:

$\bar{X} \cap X = \emptyset$ ו $\bar{X} \cup X = V$.

דוגמה:
בגרף הבא



א. נגדיר כ - X את $X=\{A\}$

אזי $\bar{X}=\{B,C,D\}$

והחתך הוא $(X,\bar{X})$.

> הגדרה: נתון גרף לא מכוון קשיר $G=(V,E)$. קשת כלשהי $e\in E$ בגרף נקראת <u>קשת חוצה</u> את החתך $(X,\bar{X})$ אם אחת מנקודות הקצה שלה ב- X והאחרת ב- $\bar{X}$.

בהמשך לדוגמה האחרונה קבוצת הקשתות החוצות את החתך

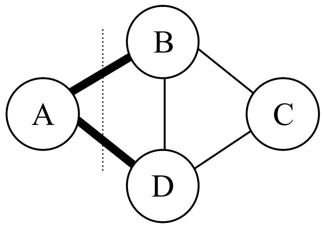
$(X,\bar{X})$ היא: $\{(A,B),(A,\cancel{D})\}$

שים לב לכך, שהקשת $(A,\cancel{C})$ לא שייכת לקבוצת הקשתות

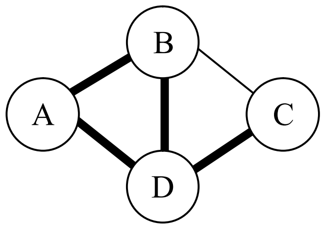
החוצות את $(X,\bar{X})$ מכיוון שקשת כזו לא קיימת בגרף הנתון. המשמעות של קבוצת קשתות חוצות את החתך היא: אם נסיר מהגרף הנתון את כל הקשתות החוצות אזי לא קיים מסלול מקודקוד כלשהו הנמצא ב – x אל קודקוד אחר נמצא ב- $\bar{x}$.

בהמשך לדוגמה, קשתות אשר שייכות לקבוצת הקשתות אשר חוצות את החתך הן מודגשות באיור שלפניך וסילוקן מהגרף תגרום שלא

יהיה מסלול מ - A ליתר הקודקודים השייכים ל- $\bar{X}=\{B,C,D\}$

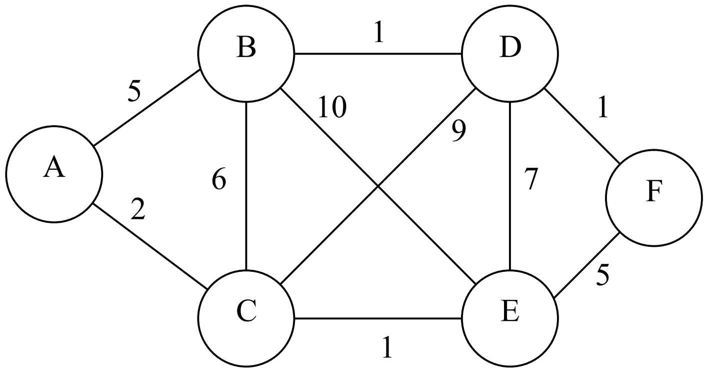


הערה : אם בגרף הנתון נגדיר את X כ- $X=\{A,D\}$ אזי
$\bar{X}=\{B,C\}$
לכן קבוצת הקשתות החוצות את החתך $(x,\bar{x})$ היא :
$\{(A,B)\ ,\ (D,B)\ ,\ (D,C)\}$
שים לב לכך, שגם כאן הקשת (A,C) לא שייכת לחתך $(x,\bar{x})$ מאותה סיבה.
לאור זאת קשתות השייכות לקבוצת הקשתות אשר חוצות את החתך $(x,\bar{x})$ הן מודגשות וסילוקן מהגרף תגרום לכך שלא יהיה מסלול מקודקוד כלשהו הנמצא ב - X (D או A) אל קודקוד אחר הנמצא ב- $\bar{x}$ (B או C).

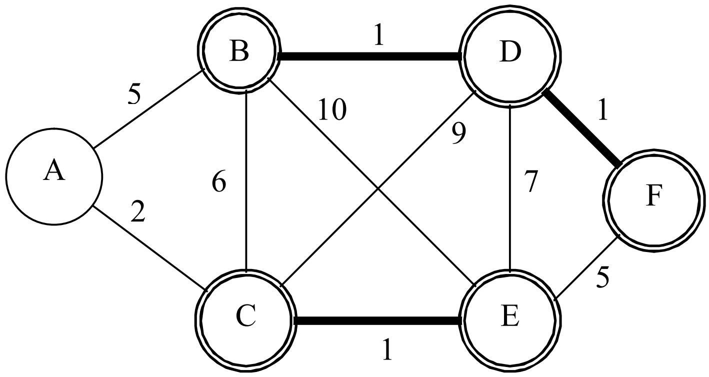


עתה נבהיר בעזרת דוגמה כיצד ניתן לזהות קשת (a,b) כך שהוספתה ל E_T, כלומר $E_T + \{ (a, b) \}$ הינה תת קבוצה של קשתות המוכלות בעץ פורש מינימלי.

בהמשך לדוגמה, למציאת עץ פורש מינימלי באמצעות אלגוריתם של קרוסקל בעבור הגרף הבא :



לאחר 3 איטרציות מקבלים תת קבוצה של קשתות,המוכלות בעץ פורש מינימלי, שהינה :



הקודקודים המוקפים בעיגול אחד שייכים ל –X, ואילו הקודקודים המוקפים בשני עיגולים שייכים ל $\bar{X}$.
בדוגמה

$\bar{X}=\{B,C,D,E,F\}$ ו $\mathbf{X=\{A\}}$

קשתות מודגשות הן:

$\{(B,D)\ ,\ (C,E)\ ,\ (D,F)\}$

הן תת קבוצה של קשתות המוכלות בעץ פורש מינימלי.
קבוצת הקשתות החוצות היא : $\{(A,B)\ ,\ (A,C)\}$
כמו כן $w(A,B)=2$ ו $w(A,C)=5$

לשים לב לכך שלא קיימת קשת בתת הקבוצה של קשתות של עץ פורש מינימלי , שחוצה את החתך $\left(X,\overline{X}\right)$.
כמו כן הקשת (A,C) היא קשת בעלת משקל הקטן ביותר מבין הקשתות החוצות את החתך $\left(X,\overline{X}\right)$ והוספתה ל - E_T שהינה תת קבוצה של קשתות המוכלות בעץ פורש מינימלי, יוצרת תת קבוצה חדשה של קשתות עץ פורש מינימלי ב – G.
כאמור באיטרציה הבאה מצרפים את הקשת (A,C) לתת קבוצה של קשתות של עץ פורש מינימלי.
לאור האמור לעיל נסיק את המשפט הבא :

<u>משפט</u>
נתון גרף לא מכוון קשיר G=(V,E) עם פונקציית משקל $W : E \to R$. יהי $E_T \subseteq E$ כך ש - E_T מוכלת בעץ פורש מינימלי כלשהו של G. יהי חתך $\left(X,\overline{X}\right)$ כך שלא קיימת קשת ב - E_T החוצה את החתך. אם קשת (a,b) כלשהי בגרף חוצה חתך ומשקלה מינימלי מבין משקליהן של כל הקשתות החוצות את החתך $\left(X,\overline{X}\right)$ אזי $E_T + \{ (a, b) \}$ גם כן מוכלת בעץ פורש מינמלי של G.

כל המעוניין יכול להוכיח את המשפט לבדו.

<u>מסקנה</u>
נתון גרף G=(V,E) בלתי מכוון וקשיר עם פונקצית משקל $\mathbf{W : E \to R}$. $E_T \subset E$ והיא מוכלת בעץ פורש מינימלי כלשהו של G. כמו כן T_1 הוא עץ (רכיב קשיר) ביער. אם הקשת (a,b) בעלת משקל מינימלי המחברת את עץ T_1 לרכיב (עץ) אחר ביער אזי $E_T + \{ (a, b) \}$ גם כן מוכלת בעץ פורש מינימלי כלשהו של G.
הוכחה
לא קיימת ב E_T קשת החוצה את החתך $\mathbf{(T1, V - T1)}$ ולכן הקשת (a,b) היא קשת הקלה עבור חתך זה, לכן לפי המשפט

הקודם $E_T + \{ (a, b) \}$ גם כן מוכלת בעץ פורש מינימלי של G.

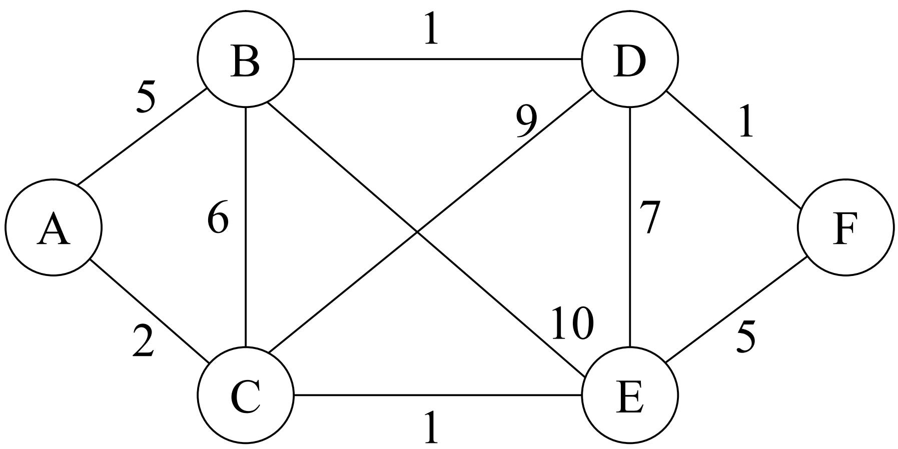
לאור האמור לעיל אנו מוכנים להציג את האלגוריתם של פרים שנכונותו הוצגה כאן.

6.4 אלגוריתם של פרים – Prim
למציאת עץ פורש מינימלי

נתונה רשת $G=(V,E)$ ומתחילים ליצור עץ פורש מקודקוד כלשהו $r \in V$ בכל שלב נוסיף קשת בעלת מישקל מינימלי המחברת את העץ לקודקוד שלא שייך לעץ. כך גדל העץ הפורש עד שהוא פורש את כל קודקודי הגרף G.
בדומה לאלגוריתם של קרוסקל גם האלגוריתם של פרים הוא אלגוריתם חמדני כי בכל צעד הוא מוסיף קשת בעלת משקל קטן ככל האפשר. ולפי המסקנה שראינו קודם אסטרטגיה כזו של בניית עץ פורש מבטיחה שבתום האלגוריתם E_T תכיל קשתות המהוות עץ פורש מינימלי.
כאמור מטרתנו למצוא עץ המכיל את כל קודקודי הגרף. לשם פיקוח על כך נשתמש בתור Q , כך ש: $V - Q$ מייצגת קבוצת הקודקודים אשר טופלו ונמצאים בעץ פורש ואילו Q מייצגת קבוצת הקודקודים שעדיין לא טופלו ולא נימצאים בעץ פורש.
מאחר שבתחילת האלגוריתם אף קודקוד של גרף נתון לא טופל כל הקודקודים יהיו בתור ובתום האלגוריתם התור Q חייב להישאר ריק, כיוון שכל הקודקודים חייבים להיות בעץ פורש.
עבור כל קודקוד v נשמור בתור Q את $K[v]$ אשר יכיל את המשקל המינימלי מבין משקלי הקשתות המחברות את הקודקוד v לקודקודים השייכים לעץ.
ברור כי בתחילת האלגוריתם $K[v] \leftarrow \infty$.
בנוסף, עבור כל קודקוד v נשמור מידע נוסף $P[v]$, שהינו ה"הורה" של v בעת בניית עץ פורש. כלומר $P[v]$ מציין את האבא של $P[v]$.
הערה: הסימון P , נבע מהסיבה ש - $P[v]$ מייצג "הורה" (Parent) של קודקוד v.

6.4.1 הצגת האלגוריתם של Prim

טרם נציג את האלגוריתם, נדגים את אופן הפעולה של האלגוריתם של Prim על הרשת הבאה:



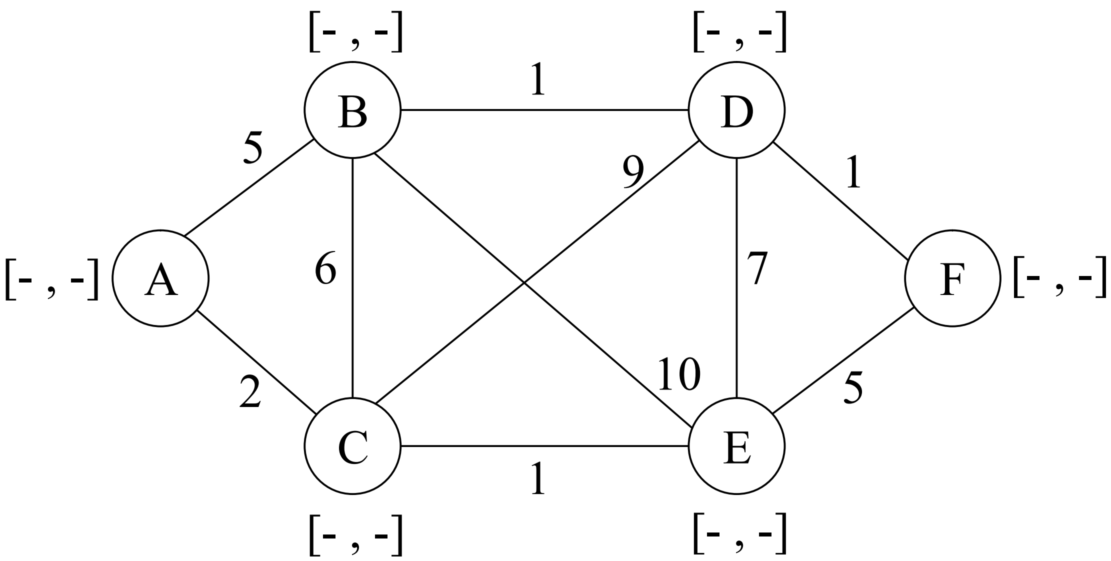
כלומר בתחילת האלגוריתם תמונת התור Q הינה:

קודקוד v	A	B	C	D	E	F
K[v]	∞	∞	∞	∞	∞	∞

בתהליך התיאור של האלגוריתם סמוך לכל קודקוד v של גרף מופיעים שני מספרים, השמאלי מייצג את הקודקוד שהינו "הורה" של v , והימני מייצג את המשקל המינימלי מבין משקלי הקשתות המחברות את הקודקוד v לקודקודים השייכים לעץ.

נתחיל את בניית העץ מקודקוד A . (בחירה של הקודקוד A נקבעה באופן שרירותי).

תמונת הרשת בהתחלה הינה:



בתחילת האלגוריתם לאף קודקוד אין "הורה".
נתחיל את בניית העץ מקודקוד A , לכן נבצע את הצעדים הבאים:

$$K[A] \rightarrow 0$$
$$P[A] \rightarrow nil$$

מאחר שהבניה של עץ פורש תחל מקודקוד A אז A יהיה שורש העץ ולכן אין לו אב ("הורה").

קודקוד v	A	B	C	D	E	F
K[v]	0	∞	∞	∞	∞	∞
P[v]	Nil	-	-	-	-	-

<u>באיטרציה ראשונה</u> נבצע את הצעדים הבאים:

א. הוצא מהתור Q את u כך ש $K[u]$ הכי קטן.
בדוגמא שלנו נוציא מהתור את הקודקוד A.

ב. בדוק את השכנים של הקודקוד u מבין הקודקודים הנמצאים בתור Q.
בדוגמה, השכנים של A הם B ו C
מכיוון ש:

$$w(A,B)=5<\infty=K[B]$$

אז נבצע :

$$P[B]\leftarrow A$$
$$K[B]\leftarrow 5$$

בנוסף , מכיוון ש

$$w(A,C)=2<\infty=K[C]$$

אז נבצע :

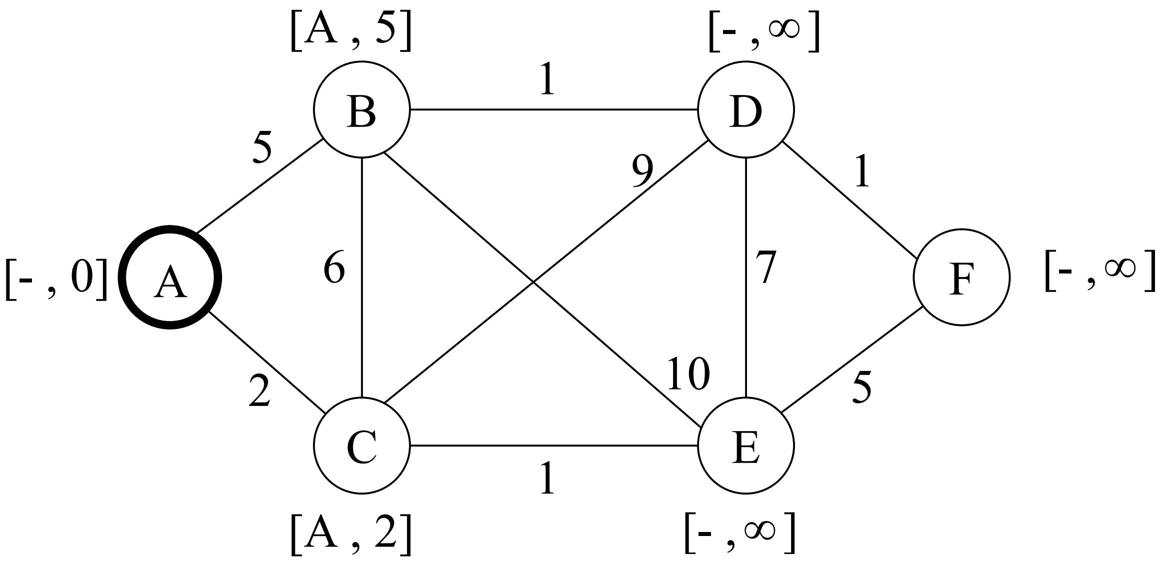
$$P[C]\leftarrow A$$
$$K[C]\leftarrow 5$$

לכן, בתום האיטרציה הראשונה מתקבל :

$Q:$

קודקוד v	B	C	D	E	F
K[v]	5	2	∞	∞	∞

תמונת ביניים הינה:



בעץ פורש יש רק קודקוד A.

<u>באיטרציה שניה</u> נבצע שוב את הצעדים הבאים:

א. הוצא מהתור Q את u כך ש K[u] הכי קטן.
בדוגמה שלנו נוציא מהתור את הקודקוד C.

ב. בדוק את השכנים של הקודקוד u מבין הקודקודים בתור Q.
בדוגמה, השכנים של הקודקוד C מבין הקודקודים נמצאים בתור Q הם B,E,D .

<u>מכיוון ש:</u>

$$w(C,B) = 6 \nleq 5 = K[B]$$

אז לא נבצע שום דבר,

<u>מכיוון ש:</u>

$$w(C,E) = 1 < \infty = K[E]$$

אז נבצע :

$$\mathbf{P[E] \leftarrow C}$$
$$\mathbf{K[E] \leftarrow 1}$$

<u>מכיוון ש:</u>

$$w(C,D) = 9 < \infty = K[D]$$

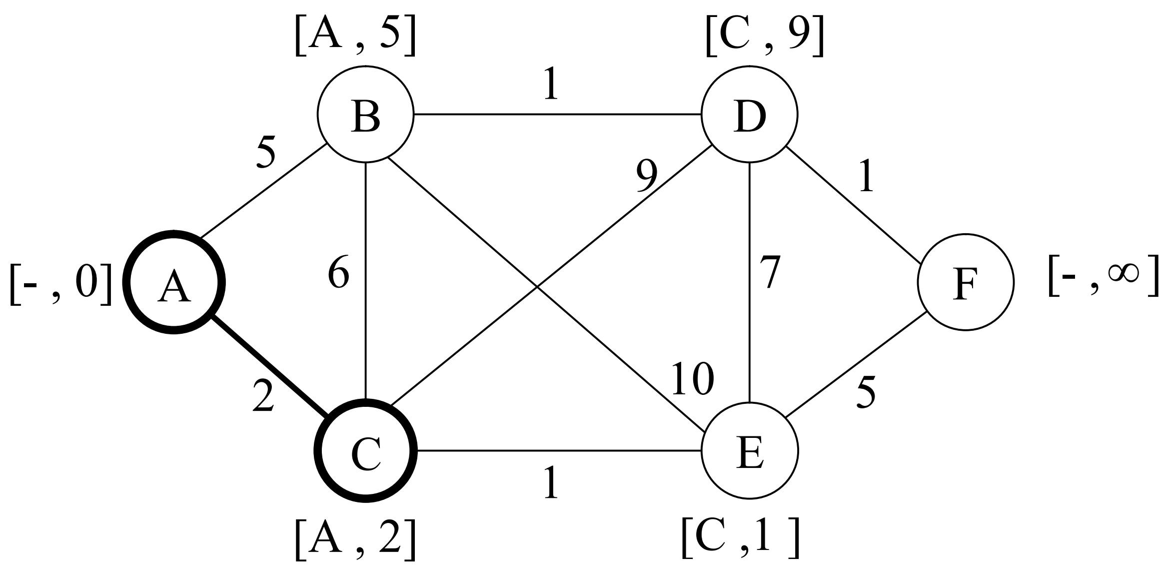
אז נבצע :

$$\mathbf{P[D] \leftarrow C}$$
$$\mathbf{K[D] \leftarrow 9}$$

לכן, בתום האיטרציה השנייה נקבל :

Q :

קודקוד v	B	D	E	F
K[v]	5	9	1	∞



הקשת (A,C) המודגשת והקודקודים {A,C} המודגשים שייכים לעץ פורש מינימלי לעתיד לבוא.

<u>באיטרציה שלישית</u> נבצע שוב את הצעדים הבאים:

א. נוציא מהתור Q את הקודקוד E מכיוון ש - K[E] הכי קטן מבין אברי התור.
ב. השכנים של הקודקוד E מבין הקודקודים הנמצאים בתור Q , הם : B,D,F .

<u>מכיוון ש:</u>

$$\mathbf{w}(E,B) = 10 \nless 5 = \mathbf{[}B\mathbf{]}$$

אז לא נבצע שום דבר.

<u>מכיוון ש:</u>

$$w(E,D) = 7 < 9 = K[D]$$

אז נבצע :

$$\mathbf{P[D] \leftarrow E}$$
$$\mathbf{K[D] \leftarrow 7}$$

<u>מכיוון ש:</u>

$$w(E,F) = 5 < \infty = K[F]$$

אז נבצע :

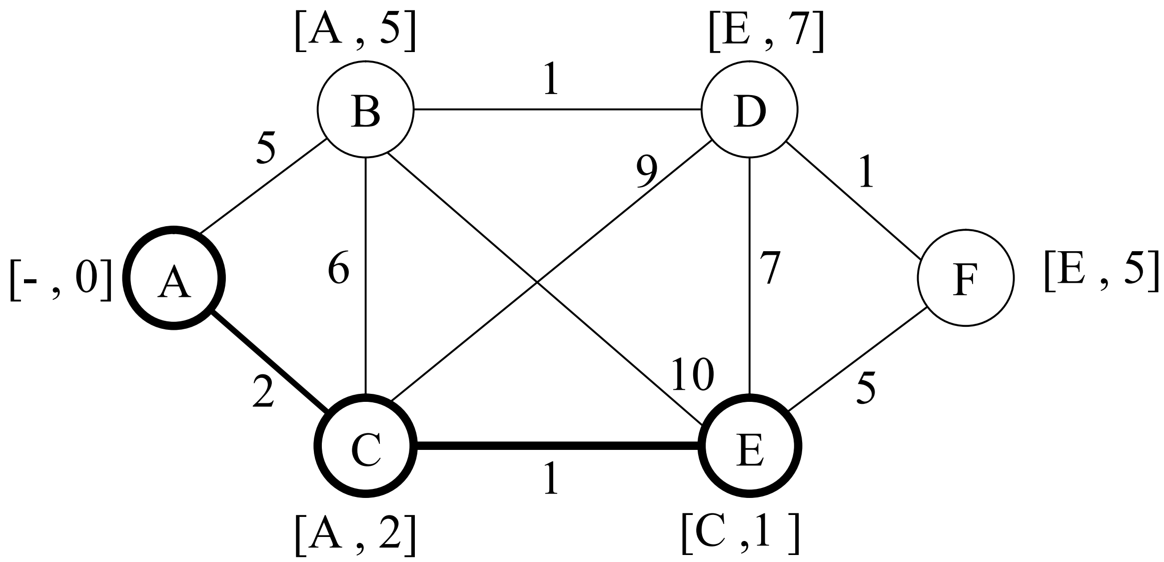
$$\mathbf{P[F] \leftarrow E}$$
$$\mathbf{K[F] \leftarrow 5}$$

לכן, בתום האיטרציה השלישית ונקבל :

Q :

קודקוד v	B	D	F
K[v]	5	7	5

תמונת ביניים הינה:



קבוצת הקשתות {(A,C) , (C,E)} המודגשות והקודקודים {A,C,E} המודגשים שייכים לעץ פורש מינימלי לעתיד לבוא.

<u>באיטרציה רביעית</u> נבצע שוב את הצעדים הבאים:

א. נוציא מהתור Q את הקודקוד u מכיוון ש K[u] הכי קטן
בשלב זה u יכול להיות קודקוד B או F.
ניבחר באופן שרירותי את הקודקוד F.

ב. השכנים של הקודקוד F , מבין הקודקודים הנמצאים בתור Q , הוא רק D .

<u>מכיוון ש:</u>

$$w(F,D) = 1 < 7 = K[D]$$

אז נבצע :

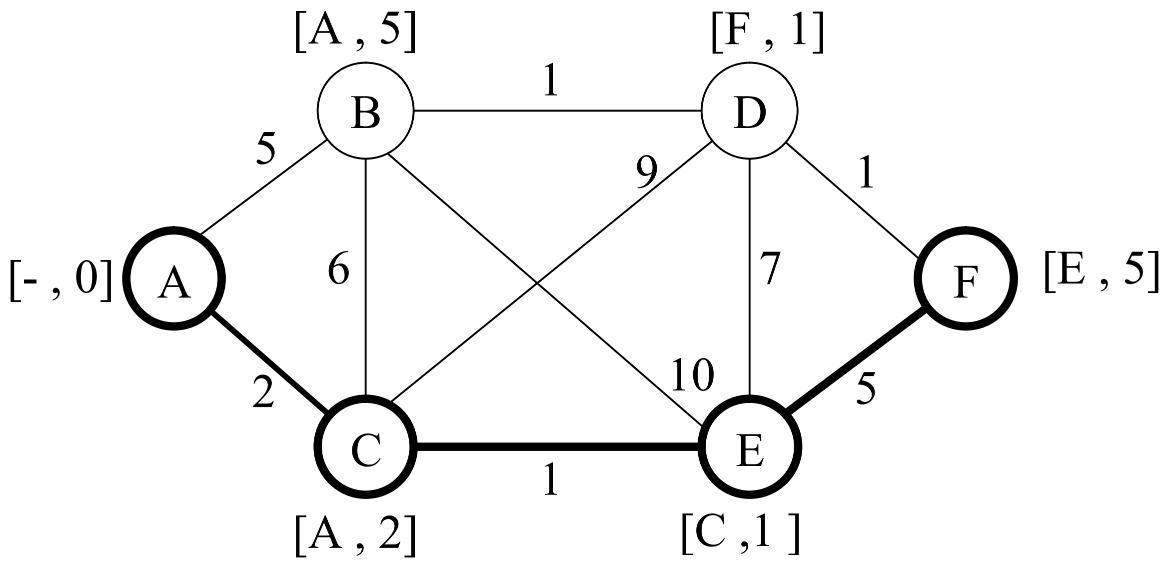
$$\mathbf{P[D] \leftarrow F}$$

$$\mathbf{K[D] \leftarrow 1}$$

לכן, בתום האיטרציה הרביעית נקבל :

Q :

קודקוד v	B	D
K[v]	5	1



קבוצת הקשתות המודגשות {(A,C) , (C,E) , (E,F)}
וקבוצת הקודקודים {A,C,E,F} המודגשים שייכים לעץ פורש מינימלי לעתיד לבוא.

<u>באיטרציה חמישית</u> נבצע שוב את הצעדים הבאים:

א. נוציא מהתור Q את הקודקוד D מכיוון ש K[D] הכי קטן מבין אברי התור.

ב. השכנים של הקודקוד D , מבין הקודקודים הנמצאים בתור Q , הוא רק B .

<u>מכיוון ש:</u>

$$w(D,B) = 1 < 5 = K[B]$$

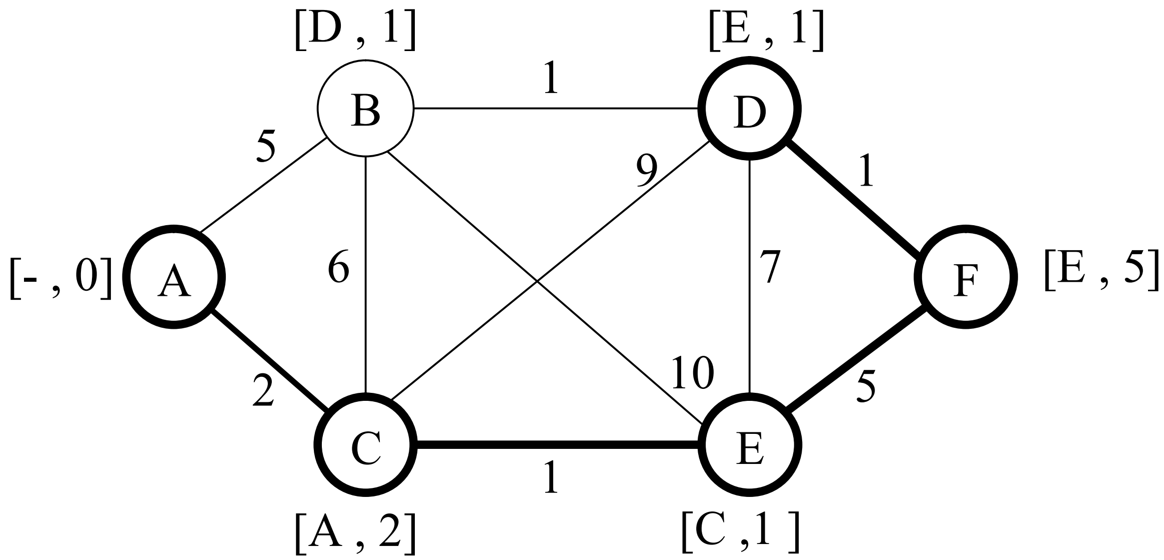
אז נבצע :

$$P[B] \leftarrow D$$
$$K[B] \leftarrow 1$$

לכן, בתום האיטרציה החמישית נקבל :

Q :

קודקוד v	B
K[v]	1



קבוצת הקשתות המודגשות {(A,C) , (C,E) , (E,F) , (F,D)}
וקבוצת הקודקודים {A,C,E,F,D} המודגשים שייכים לעץ פורש מינימלי לעתיד לבוא.

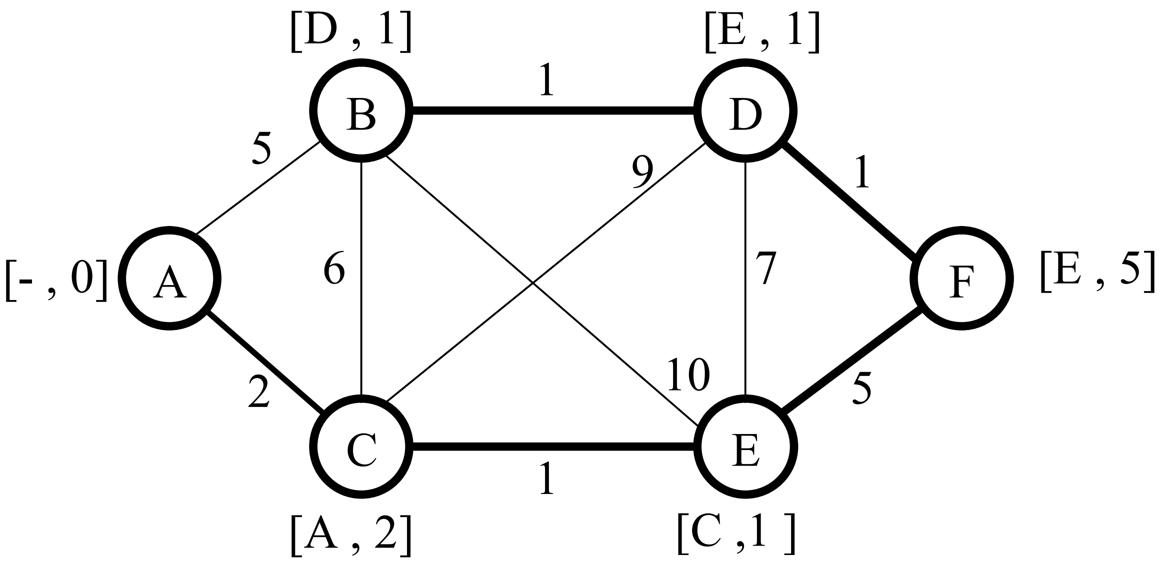
<u>באיטרציה שישית</u>

מאחר והתור Q לא ריק נבצע את הצעדים הבאים:

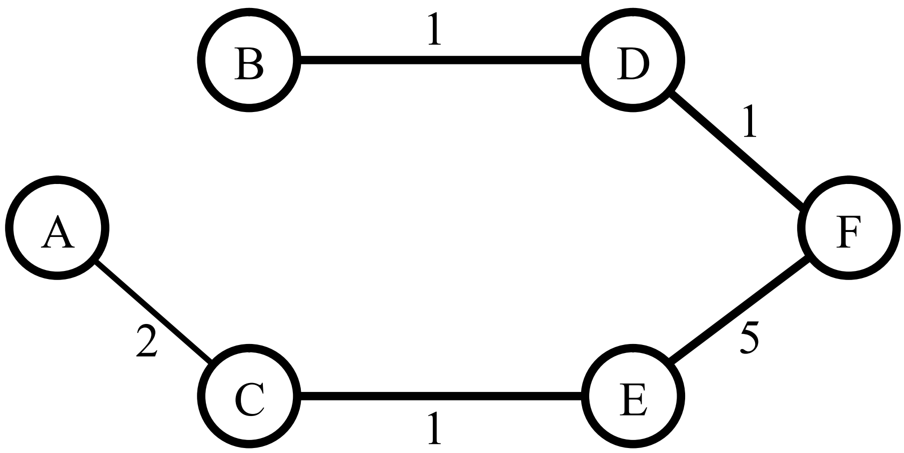
א. נוציא מהתור Q את הקודקוד B.

ב. כעת נבדוק את הקודקודים השכנים של קודקוד B, הנמצאים עדיין בתור Q. בשלב זה אין קודקודים כאלו ולכן אין במה לטפל.

ולכן תוצאת המצב שנתקבל עד כה מוצגות בטבלה ובתרשים הבא:
התור Q <u>הינו</u> ריק.



מאחר שהתור ריק האלגוריתם הסתיים ועץ פורש מינימלי שנתקבל סופית הינו :



עתה נוכל לסכם את האלגוריתם כדלהלן:

אלגוריתם Prim (G , r)

האלגוריתם מקבל כקלט את הגרף הבלתי מכוון וקשיר G ואת השורש r של העץ הפורש המינימלי שהאלגוריתם צריך לבנות ולהחזיר.

צעד 1.

לכל קודקוד v בצע :

1.1 $K[v] \leftarrow \infty$

1.2 הכנס את K[v] לתור Q.

סוף לולאה.

צעד 2.

2.1 $K[r] \leftarrow 0$

2.2 $P[r] \leftarrow nil$

צעד 3.

כל עוד התור Q לא ריק (כלומר לא כל הקודקודים נמצאים עדיין בעץ פורש T) בצע :

3.1 הוצא קודקוד u מהתור Q , כך שלכל קודקוד $v \in Q$ $K[u] = \min_v \{K[v]\}$ (כלומר מוציאים מהתור Q קודקוד u כאשר $K[v] > K[u]$ לכל קודקוד v בתור).

3.2 עבור כל קודקוד v שהינו שכן של קודקוד u ועדיין נמצא בתור Q בדוק:

אם $w(u,v) < K[v]$ (כלומר הקשת הנוכחית (u, v) בעלת משקל יותר קטן מבין המשקולות שעל הקשתות שבחרנו עד כה ונגעו ב - v)

אז בצע :

3.2.1 $P[v] \leftarrow u$ (* נקבע הורה חדש *)

3.2.2 $K[v] \leftarrow w(u,v)$ (* נקבע משקל חדש *)

סוף לולאה.

סוף לולאה.

בכל שלב של אלגוריתם <u>פרים</u> אנו מקבלים תור של קודקודי הגרף וממנו מוציאים קודקוד אחד u , בעל הערך K[u] הקטן ביותר ובאמצעותו מנסים לעדכן עבור כל קודקוד v , הנמצאים בתור , את ערכיהם K[v] ולהכניס לתור בחזרה את הערכים המעודכנים.
לכן יש צורך לייצג את התור באמצעות מבני נתונים שעל מבנה זה מוגדר מספר פעולות בסיסיות הבאות:

EXTRACT_MIN(Q) – מוציאה מהתור Q את האיבר u בעל הערך (K[u]) הקטן ביותר מבין כל האיברים שבתור Q ומחזירה אותו.
DECREASE_KEY(Q, v, k[v]) – מכניסה איבר v , עם הערך החדש K[v] , לתוך Q.
לאור האמור לעיל לפניך אלגוריתם של Prim תוך שימוש בפעולות הבסיסיות המוגדרות על טיפוס נתון תור.

<u>PRIM(G,r)</u>

```
step 1:   //INIT
          for each vertex v do
                K[v]  ←  ∞
                P[v]  ←  NULL
step 2:   K[r]  ←  0
          P[r]  ←  NULL
          PQ  ←  V   //Priority Queue holds the
                     //vertices outside the tree
                //GROW TREE
step 3:   while PQ ≠ Φ do
            (3.1) u ← EXTRACT_MIN(PQ)
            (3.2) for each v ∈ adj[u] do
                  (3.2.1) if v ∈ PQ and w(u,v) < K[v] then
                          (3.2.1.1) K[v] ← w(u,v)
                          (3.2.1.2) P[v] ← u
                          (3.2.1.3) DECREASE_KEY(PQ, v, K[v])
```

הערה:
adj [u] מציין קבוצת קודקודים שהם שכנים של קודקוד u.

6.4.2 יעילות אלגוריתם שלPrim

מאחר שבכל איטרציה של האלגוריתם אנו מוצאים מהתור Q קודקוד u בעל ערך K[u] מינימלי, נחליט לממש את התור שהינו תור עם עדיפות כערמה של עץ בינרי.
אי לכך:

step 1: מספר הצעדים לבניית הערמה (בניית התור) הוא $O(|V|)$

step 2: חישוב K[r] דורש זמן $O(1)$
חישוב P[r] דורש זמן $O(1)$
המשפט $PQ \leftarrow V$ דורש זמן $O(|V|)$
לכן סך הכל הזמן הנדרש בצעד זה הוא $O(|V|)$.

step 3:
באיטרציה כלשהי i:

step 3.1: הוצאת איבר מינימלי מהערמה לוקחת זמן $O(\log|V|)$.

step 3.2: מתבצע adj[u] פעמים (כמספר הקודקודים הקשורים לקודקוד u_i או כאורכה של הרשימה המקושרת המייצגת רשימת הסמיכות של הקודקוד u).

step 3.2.1: כל קודקוד יכול להמצא בתור או מחוץ לתור. לשם פיקוח על כך נשתמש במערך בוליאני Exist כך ש Exist[v] מציין אם קודקוד v נמצא בתור או לא.
אי לכך הזמן הנדרש לפעולה הבודקת האם קודקוד מסויים נמצא בתור או לא, הוא $O(1)$.

step 3.2.1.1: דורש זמן $O(1)$.

step 3.2.2.2: דורש זמן $O(1)$.

step 3.2.2.3: הכנסת ערך חדש לערמה דורשת זמן $O(\log|V|)$.

יש לשים לב שמטפלים בקודקוד u_i וברשימת הסמיכות שלו רק פעם אחת לכן בצעד 3 בהיטרציה כלשהי i הזמן הנדרש הוא

$$O(\log|V|) + adj[u]\cdot[O(1) + O(\log|V|]$$
$$\cong O(\log|V|) + adj[u]\cdot O(\log|V|)$$

מאחר שהצעד 3.2 מתבצע $|v|$ פעמים אז:
סך הכל מספר הצעדים בכל האיטרציות הוא :

$$\sum_{u\in V} adj[u]\cdot O(\log|V|) = O(\log|V|)\cdot \sum_{u\in V} adj[u]$$

מחד $\sum_{u\in V} adj[u]$ הוא הסכום של אורכי כל הרשימות הסמיכות בעבור כל קודקודי הגרף הנתון, מאידך ידוע כי בגרף לא מכוון מתקיים: $\sum_{u\in V} adj[u] = 2|E|$.
לכן צעד 3.2 דורש זמן $O(|E|\log|V|)$.
מאחר שצעד 3.1 מתבצע $|V|$ פעמים ובכל איטרציה הזמן הנדרש הוא $(\log|V|)$ אזי סך הכל זמן הריצה של הצעד 3.1 הוא $O(|V|\log|V|)$.
לכן סך הכל הזמן הנדרש לצעד 3 הוא:

$$O(|V|\log|V|) + O(|E|\log|V|)$$
$$\approx O(|E|\log|V|)$$

סופית סיבוכיות זמן הריצה של האלגוריתם הנתון הינה :

$$O(\max(|V|, |V|, |V|\log|V|, |E|\log|V|)) = O(|E|\log|V|)$$

זמן ריצה של אלגוריתם קרוסקל הינו: $\mathbf{O(|E|\log|E|)}$
ושל Prim: $O(|E|\log|V|)$
אך $|\mathbf{E}| \le \mathbf{n}^2$
ו $|V| \le n$
לכן $O(\log|E|) \approx O(\log|V|)$

לכן זמני הריצה של קרוסקל ושל פרים אסימפטוטית זהים.
הערה: אם נממש את התור Q בעזרת מבני נתונים מתקדם הנקרא "ערמות פיבונצ'י" שבאמצעותו ניתן לממש את הצעד 3.2.1.3 בזמן $O(1)$. לכן סיבוכיות זמן הריצה של אלגוריתם Prim הינו: $O(|V|\log|V|+|E|)$.

<u>הערות חשובות</u>

1. משקלי הקשתות של הגרף הנתון $G=(V,E)$ אינם חייבים להיות חיוביים, מהסיבה הבאה:

   א. קודם נגדיר את :

$$Inc = \min \{ w(e) \mid e \in E \text{ וגם } w(e) < 0 \}$$

   Inc מציין את המשקל השלילי הקטן ביותר מבין משקלי הקשתות של הגרף הנתון.

   ב. אחרי כן נגדיר גרף חדש $\mathbf{G_1 = (V, E_1)}$ שבו לכל קשת $\mathbf{e \in E}$ נצרף ל-E_1 קשת שמשקלה $\mathbf{e - Inc = e + (-Inc)}$. מאחר שInc הוא בעל ערך שלילי אזי (-Inc) הוא בעל ערך חיובי. הוספת קבוע (-Inc) לכל המשקלים הופכת את משקלי כל הקשתות לחיוביים.
   לדוגמה אם משקלי הקשתות הם: 5, 2, 0, -7, -8, -10
   אזי ברור כי $Inc = -10$ אך $-Inc = -(-10) = 10$.

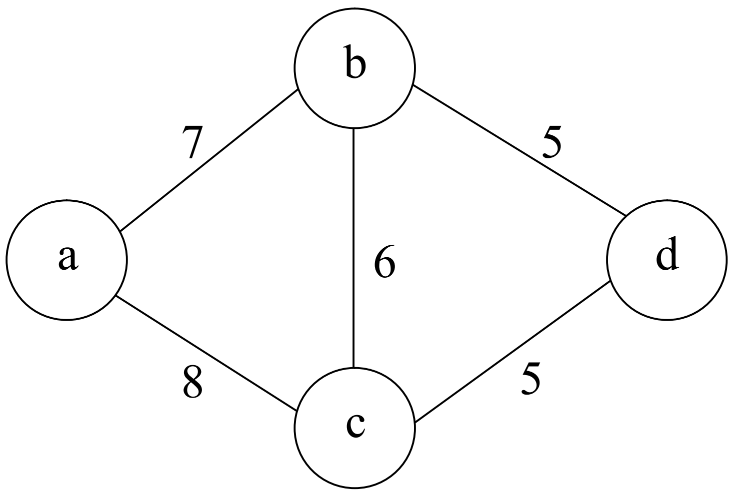
   ועל ידי הוספת 10 למשקלי קשתות הופכים את משקלי הקשתות לחיוביים ואלו הם: 15, 12, 10, 3, 2, 0
   אך ברור כי הוספת מספר חיובי קבוע לכל המשקלים ועל ידי כך הפיכתם לחיוביים, לא תשנה את הפיתרון האופטימלי.

2. ניתן למצוא עץ פורש מינימלי בעזרת האלגוריתם הפשוט הבא:

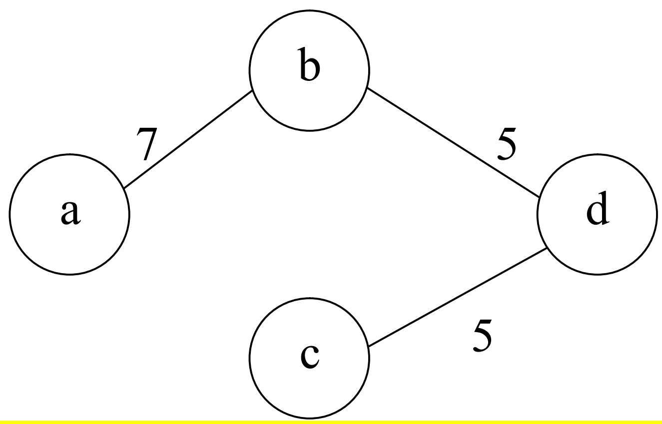
   א. חזור על התהליך הבא:

מצא קשת בעלת המשקל הגדול ביותר בגרף הנתון
והסר אותו מהגרף, בתנאי שהגרף נשאר קשיר.
ב. בתום שלב א' העץ הפורש שמתקבל הינו עץ פורש
מינימלי.

לדוגמה: בעבור הרשת הבאה:



עץ פורש מינימלי שמתקבל הוא :



3. כל אלגוריתמים למציאת עץ פורש מינימלי ניתנים
להמרה למציאת עץ פורש מקסימלי. לפניך 2 גירסאות
של שני אלגוריתמים אשר מוצאים עץ פורש מקסימלי.

<u>גירסה I :</u>

- חזור על התהליך הבא :
  מצא קשת בעלת המשקל <u>הקטן</u> ביותר בגרף הנתון
  והסר אותו מהגרף בתנאי שהגרף נשאר קשיר.
- עתה העץ הפורש שמתקבל הוא עץ פורש
  מקסימלי.

גירסה II :

עץ פורש מקסימלי ניתן לבנות גם כך: בהתחלה בוחרים את הקשת בעלת העלות המקסימלית, לאחר מכן את זו בעלת העלות הגדולה ביותר מבין הקשתות הנותרות וכך הלאה ובלבד שבכל שלב הקשת שנבחרה לא סוגרת (לא יוצרת) מעגל.

4. נתון גרף קשיר ולא מכוון $G=(V,E)$, עם פונקציית המשקל $w: E \to R$. נתונה פונקציה מונוטונית עולה ממש $f(x)$ (לכל $x<y \Rightarrow f(x)<f(y)$). נגדיר פונקצית משקל חדשה כך: $w'(x) = f(w(x))$. כל עץ פורש מינימלי לפי w נשאר עם עץ פורש מינימלי לפי w'.
נימוק:
כזכור באלגוריתם של קרוסקל קודם ממיינים את משקלי הקשתות מכיוון שהאלגוריתם מתבסס על יחס הסדר בין המשקלים של הקשתות.
אך לפי ההגדרה פונקציה מונוטונית עולה שומרת על אותו הסדר, כי אם בסדרה הממוינת המקורית
$w(x)<w(y)$ אזי
$w'(x) = f(w(x)) < f(w(y)) = w'(y)$
לכן אם נגדיר את פונקצית המשקל החדשה על סדרה ממויינת של משקלי הקשתות של הגרף הנתון נקבל סדרה חדשה של משקלים ממויינים לפי w' מכיוון ש – $f(x)$ שומרת על יחס הסדר.

לכן האלגוריתם של קרוסקל ימצא עץ פורש מינימלי לפי w' הזהה לעץ פורש מינימלי לפי w .

5. נתונים שני עצים פורשים מינימלים T_1, T_2 של גרף נתון $G=(V,E)$ עם פונקציית משקל $w: E \to R$. אם נסדר לכל עץ את סדרת משקלותיו בסדר עולה אזי הסדרות הן זהות.
נימוק:

נסמן ב – $W(T_1) = W(T_2) = M$

נניח שסדרות המשקלות הן:

$\mathbf{a_1 \le a_2 \le a_3 \le ...a_i \le a_n}$ בעבור T_1

$\mathbf{b_1 \le b_2 \le b_3 \le ...b_i \le b_n}$ בעבור T_2

עתה נניח ש- i הוא האינדקס הראשון בו $\mathbf{a_i < b_i}$.
נפעיל על קשתות הגרף את הפונקציה המונוטונית הבאה (שהיא פונקציה עולה):

$$f(x) = \begin{cases} x & X \le a_i \\ x+1 & X > a_i \end{cases}$$

$$\mathbf{w'(T_1) = f(w(T_1) = w(T_1) + 1 \cdot (n-i) = M + (n-i)}$$
$$\mathbf{w'(T_2) = f(w(T_2) = w(T_2) + 1 \cdot (n-i+1) = M + (n-i+1)}$$

כלומר $\mathbf{W'(T_1) < W'(T_2)}$
בסתירה לטענה הקודמת הטוענת שכל עץ פורש מינימלי לפי w נשאר גם עץ פורש מינימלי לפי w' כאשר $w'(x) = f(w(x))$ בעבור פונקציה f מונוטונית עולה. כלומר, מכיוון ש $f(x)$ פונקציה מונוטונית עולה, היינו צריכים לקבל ש $\mathbf{W'(T_1) = W'(T_2)}$. אך בגלל שקיבלנו $\mathbf{W'(T_1) < W'(T_2)}$
זוהי סתירה להנחה ולכן לא קיים אינדקס i כזה שבו $a_i < b_i$ ולכן לכל i $a_i = b_i$ כלומר שתי הסדרות זהות.

6. אפשר למצוא עץ פורש מינימלי עם עדיפות לקשתות, כך למשל נתון גרף קשיר לא מכוון $G=(V,E)$ עם פונקציית המשקל $\mathbf{w: E \to R}$. חלק מהקשתות צבועות בכחול והיתר צבועות בלבן. אנו רוצים למצוא עץ פורש מינימלי עם מספר מקסימלי של קשתות כחולות. כאמור באלגוריתם של קרוסקל ראשית כל ממיינים את קשתות הגרף לפי סדר עולה. להשגת המטרה, בעת המיון כאשר ישנם קשתות בעלות אותו משקל, בסידרה הממויינת של משקלי הקשתות ניתן עדיפות לקשתות הכחולות, כלומר קודם

נרשום בסדרה הממויינת את הקשת הכחולה ולאחריה קשת לבנה שיש לה משקל זהה כמשקלו של הקשת הכחולה.

נציג את הפתרון לבעיה הנתונה בדרך נוספת.

- נסמן ב – H את ההפרש המינימלי בין כל שני משקלי הקשתות שבגרף הנתון.
- נגדיר $\varepsilon < \frac{H}{|V|-1}$
- נוריד ממשקלו של כל קשת כחולה את הגודל ε וכך נקבל פונקציית משקל חדשה 'w. בעבור קשתות בעלות אותו משקל, פעולה זו נתנה עדיפות לקשתות הכחולות על פני קשתות הלבנות בכך שהורדנו מהן את הגודל ε.
- בשלב זה נפעיל את האלגוריתם של קרוסקל למציאת עץ פורש מינימלי על פי 'w.

ברור כי ~~ש~~לכל עץ פורש T מתקיים:

$$W(T)-(|V|-1)\cdot\varepsilon \le W'(T) \le W(T)$$

עתה נראה שאם T_1 ו T_2 שני עצים פורשים שעבורם מתקיים

$W(T_1) < W(T_2)$ אזי $W'(T_1) < W'(T_2)$

$$W'(T_1) \le W(T_1) \le W(T_2) - H = W(T_2) - (|V|-1)\cdot\frac{H}{|V|-1}$$

כלומר

$$W'(T_1) \le W(T_2) - (|V|-1)\cdot\varepsilon \le W'(T_2)$$

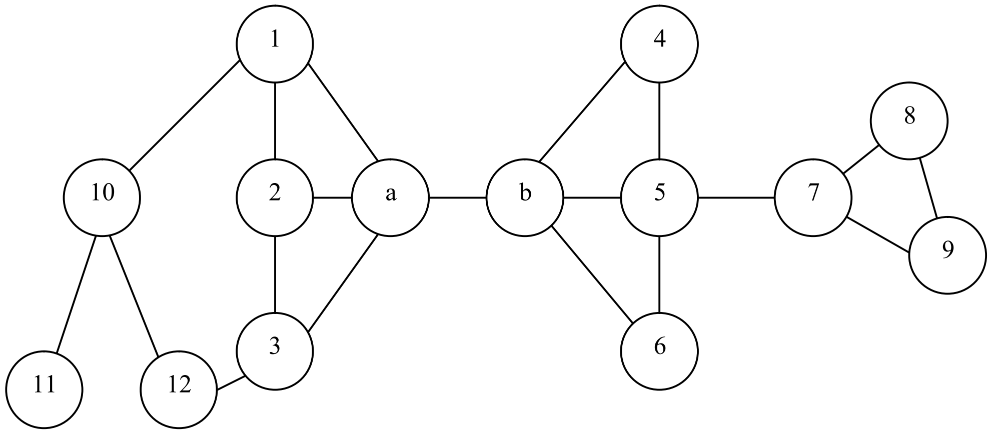
כלומר שקיבלנו

$$W'(T_1) < W'(T_2)$$

לכן הראינו שלעץ פורש מינימלי לפי W הוא גם עץ פורש מינימלי לפי 'W. מבין כל עצים פורשים מינימלים לפי W , W' תיתן עדיפות לאלה עם מספר מקסימלי של קשתות כחולות לכן עץ פורש מינימלי לפי W' הוא עץ פורש מינימלי לפי W עם מקסימום קשתות כחולות.

שאלות

1. הוכח או הפרך את הטענה הבאה : אם בגרף פשוט כלשהו G דרגתו של כל קודקוד היא לפחות 2 אזי בגרף G אין קשת מפרידה.
2. הוכח או הפרך את הטענה הבאה: אם בגרף פשוט כלשהו G דרגתו של כל קודקוד זוגית אזי אין קשת מפרידה.
3. זהה בגרף הבא את כל הקשתות המפרידות.



.4

נתון גרף לא מכוון $G=(V,E)$, ופונקציית משקל $W:E\rightarrow R^{+}$ כל קשת בגרף צבועה בשחור או בלבן. בהנתן עץ פורש T נסמן ב - m_1 את מספר קשתותיו השחורות, וב - m_2 את מספר קשתותיו הלבנות. המטרה היא למצוא, מבין העצים הפורשים המינימלים, את זה שעבורו $|m_1-m_2|$ מקסימלי.

א. תן אלגוריתם מהיר ככל שתוכל לבעיה.
ב. מה סיבוכיות זמן הריצה של האלגוריתם ? הסבר.

.5

נתון גרף לא מכוון $G=(V,E)$ עם צבע $c(e)$ (לבן או שחור) לכל קשת e ומשקל שלם $1\leq w(e)\leq 100$ לכל קשת e, כאשר הגרף מיוצג על ידי רשימות שכנות ומשקל כל קשת וצבעה מופיעים ליד הופעותיה ברשימות השכנות.

א. כתוב אלגוריתם יעיל המוצא מבין כל העצים הפורשים של G המכילים את המספר הגדול ביותר האפשרי של קשתות לבנות, עץ כזה בעל משקל מינימלי.
ב. מהי סיבוכיות זמן הריצה של האלגוריתם ?

.6

נתון גרף $G=(V,E)$ לא מכוון , קשיר ועם פונקצית משקל הבאה: $W:E\rightarrow\{1,3,10\}$ לכל קשת $e\in E$, כאשר הגרף מיוצג על ידי רשימות שכנות.

א. כתוב אלגוריתם יעיל המוצא קבוצת קשתות המכילה לפחות קשת אחת מכל מעגל בגרף ומשקלה הכולל מינימלי.
ב. מהי סיבוכיות זמן הריצה של האלגוריתם שהיצעת ?

.7

נתון גרף לא מכוון, ממושקל ומיוצג על ידי רשימות שכנות כשמשקל כל קשת ליד הופעתה ונתונה קשת מסויימת e בגרף.

א. כתוב אלגוריתם יעיל הקובע האם יש בגרף עץ פורש מינימלי המכיל את הקשת המסויימת e.

ב. מהי סיבוכיות זמן הריצה של האלגוריתם שהיצעת ?

.8

נתון גרף $G=(V,E)$ לא מכוון וקשיר ובו לכל קשת צבע שחור או לבן.

א. כתוב אלגוריתם יעיל ככל האפשר המוצא בגרף עץ פורש בעל מספר מירבי של קשתות לבנות.

ב. מהי סיבוכיות זמן הריצה של האלגוריתם שהיצעת ?

.9

נתון גרף קשיר לא מכוון $G=(V,E)$ ומשקולות חיוביים שלמים על הקשתות.

כתוב אלגוריתם המוצא עץ פורש שהחזקה הגבוה ביותר של 2 המחלקת את מכפלת משקלים של קשתות העץ היא מקסימלית.

.10

נתון גרף לא מכוון $G=(V,E)$ עם פונקציית משקלים על הקשתות, וידוע שכל המשקלים על הקשתות שונים זה מזה.

הוכיחו שקיים עץ פורש מינימלי יחיד בגרף.