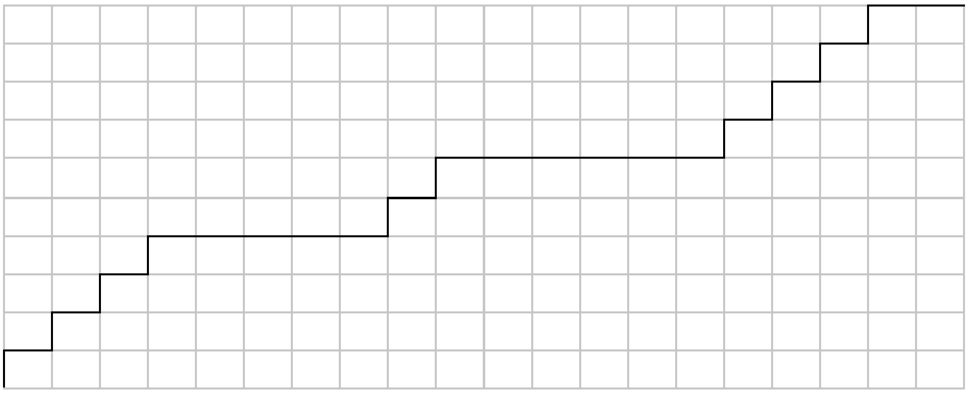
חידה לחימום

קנגורו מנתר במעלה גרם מדרגות, אשר מחולק לקטעים שביניהם משטחים ישרים, דוגמת האיור הבא:



הקנגורו מנתר 1, 2 או 3 מדרגות בניתור. בהגיעו למשטח ישר הוא מנתר מקצהו האחד לקצהו השני בניתור אחד (עבור כל אורך של משטח).

עליכם לכתוב תכנית אשר הקלט שלה הוא תיאור גרם המדרגות. בקלט נתון תחילה מספר שלם N המקיים $10 \leq N \leq 100$, ואחריו נתונים N זוגות של מספרים שלמים חיוביים, כאשר המספר הראשון בכל זוג מתאר מספר מדרגות רצופות עולות והמספר השני – אורך של משטח ישר בהמשך אותן מדרגות (ולפני רצף המדרגות העולות הבא).

למשל, הקלט לאיור הוא: 3 4 5 2 6 4 2 (משמאל לימין. ה-3 השמאלי, שהוא הנתון הראשון, מציין את מספר זוגות הנתונים המתארים את מעלה המדרגות).

הפלט הוא מספר הדרכים השונות לטיפוס במעלה המדרגות, כאשר שתי דרכים נחשבות שונות אם סדרות גדלי הקפיצות שבהן, במעלי המדרגות, הן שונות.

לדוגמה:

דרך אחת לטיפוס היא: 1, 2, 1, ramp, 1, 1, ramp, 1, 3, ramp

דרך אחרת לטיפוס היא: 1, 1, 2, ramp, 1, 1, ramp, 1, 3, ramp

דרך נוספת היא: 1, 1, 2, ramp, 2, ramp, 1, 3, ramp

וישנן דרכים רבות נוספות.

מבני נתונים ויעילות אלגוריתמים

(15.2.2015)

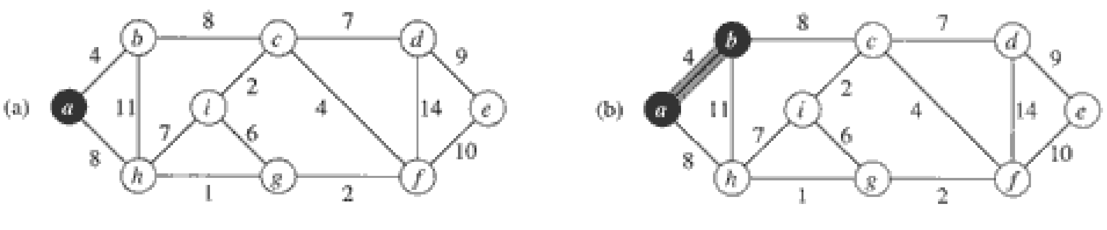
1. האלגוריתם של פרים (Prim) למציאת עץ פורש מינימלי

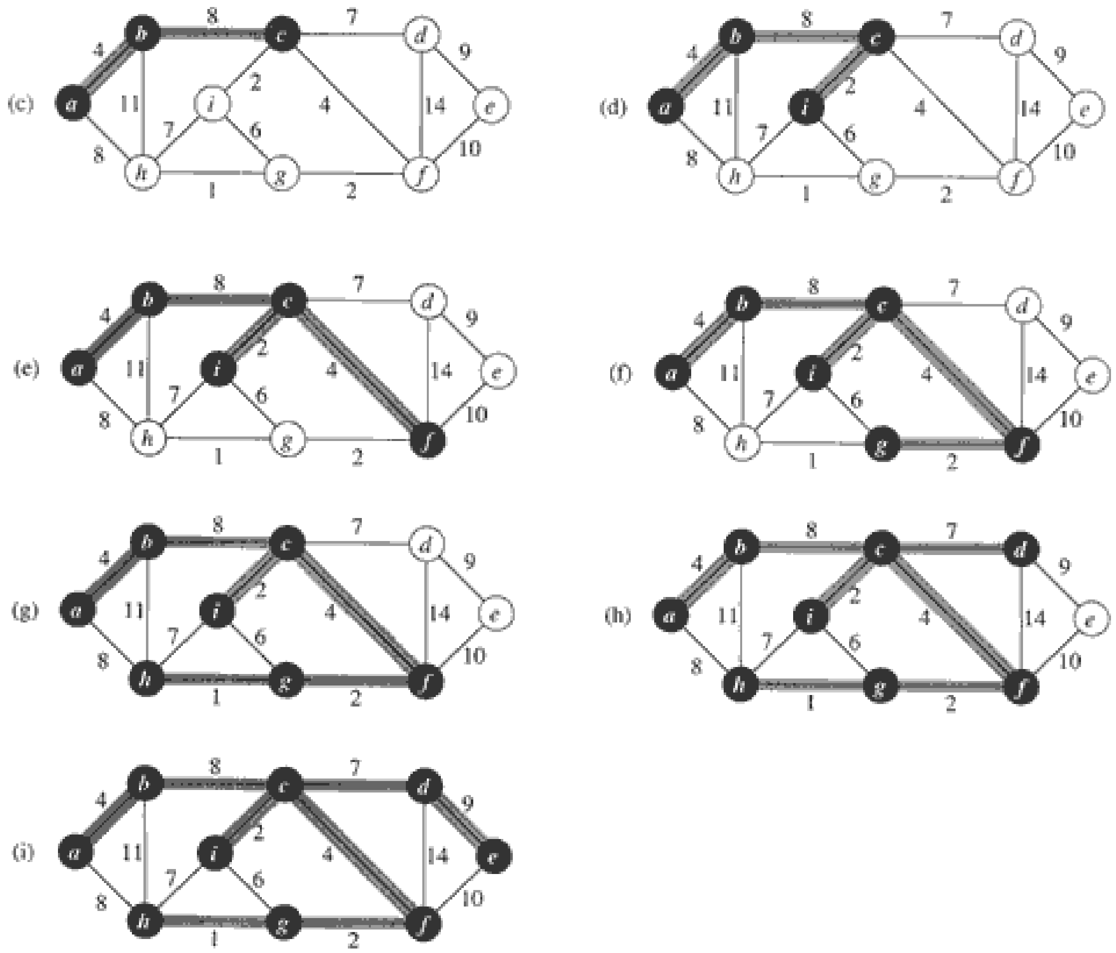
האלגוריתם מתחיל מקודקוד שרירותי r והוא בונה עץ שהולך וגדל עד שהוא פורש את כל קודקודי V. בכל צעד מוסיפים לעץ A קשת קלה המחברת את A עם קודקוד שאינו משתייך לעץ.

האלגוריתם מקבל כקלט את הגרף הקשיר G ואת השורש r של העץ הפורש המינימלי שהאלגוריתם צריך לבנות. במהלך ביצוע האלגוריתם, כל הקודקודים אשר אינם שייכים לעץ נשמרים בתור קדימויות Q המבוסס על שדה מפתח בשם key. עבור כל קודקוד v, התכונה v.key מכילה את המשקל המינימלי מבין משקלי הקשתות המחברות את הקודקוד v לקודקודים השייכים לעץ. אם לא קיימת קשת כזו, אז $v.key = \infty$.

```
PRIM(G,r)
1     Q ← V
2     for each vertex u ∈ V
3        do u.key ← ∞
4     r.key ← 0
5     r.previous ← NULL
6     while Q ≠ ∅
7        do u ← EXTRACT-MIN(Q)
8           for each vertex v adjacent to u
9              do if v ∈ Q and w(u,v) < v.key
10                   then v.previous ← u
11                        v.key ← w(u,v)
```

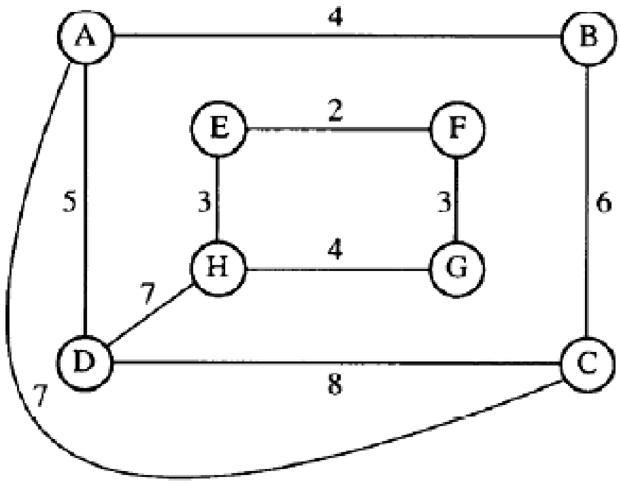
ביצוע האלגוריתם של פרים תלוי בדרך שבה מממשים את תור הקדימויות Q. אם נממש אותו בתור ערימת מינימום (minimum heap) אז פעולות האתחול בשורות 1-5 יתבצעו בזמן $\Theta(|V|)$. הלולאה שבשורה 6 תתבצע $|V|$ פעמים, ומאחר שכל פעולת EXTRACT-MIN מתבצעת בזמן $\Theta(\log|V|)$, נקבל שהזמן הכולל לביצוע כל הקריאות לפעולה זו הוא $\Theta(|V|\log|V|)$ (האמת היא, שאומנם זו הסיבוכיות הנכונה, אולם הנימוק אינו מוצלח. מדוע?). לולאת ה-for שבשורות 8-11 תתבצע $\Theta(|E|)$ פעמים בסך הכול, שכן סכום אורכי כל רשימות הסמיכות הוא $2|E|$. בתוך לולאת ה-for, בדיקת השייכות ל-Q שבשורה 9 ניתנת למימוש בזמן קבוע על-ידי כך שנשמור עבור כל קודקוד דגל בוליאני המציין אם הוא נמצא או אינו נמצא ב-Q, ונעדכן דגל זה כאשר הקודקוד נמחק מ-Q. בשורות 10-11 משנים את המפתח של האיבר v שבתוך Q, וזה ייקח $\Theta(\log|V|)$. זמן הריצה הכולל, לפיכך, הוא $\Theta(E\log V + V\log V) = \Theta(E\log V)$.



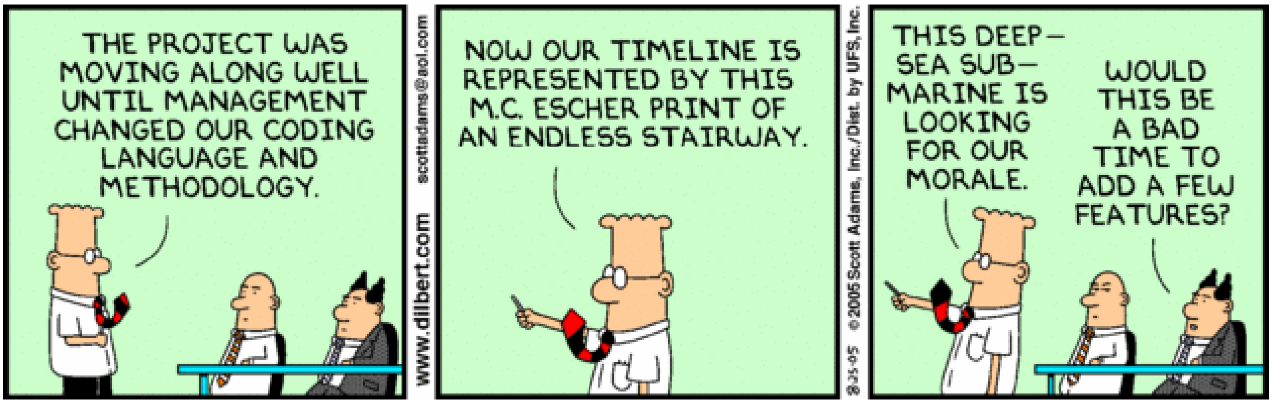


<u>שאלה (ממבחן של משרד החינוך)</u>

באיור לשאלה נתונה רשת.

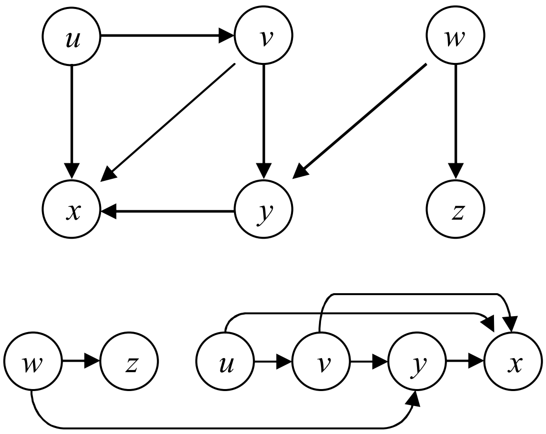


מצא לרשת הנתונה, באמצעות אלגוריתם Prim , עץ פורש מינימלי המתחיל בקדקוד A
תאר במחברת הבחינה את העץ באופן סכמתי.



2. מיון טופולוגי (Topological Sort)

בהינתן גרף מכוון ללא מעגלים (גמ"ל; ראשי תיבות באנגלית – Directed Acyclic Graph – DAG) אפשר לבצע עליו מיון טופולוגי (Topological Sort).

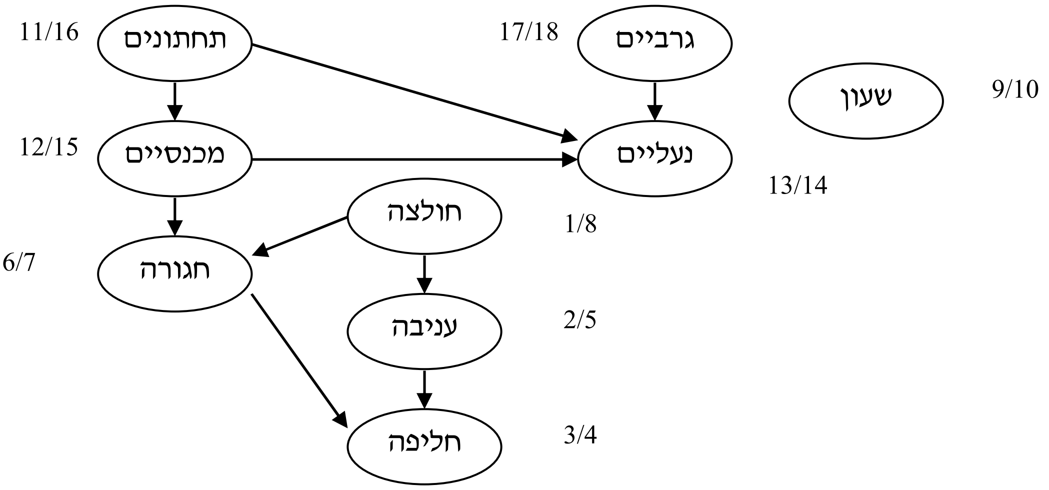


מיון טופולוגי של גמ"ל $G = (V,E)$ הוא סידור לינארי של קבוצת הקודקודים V, כך שלכל קשת $(u,v) \in E$, הקודקוד u יופיע לפני הקודקוד v בסידור הזה. ניתן לראות מיון טופולוגי של גרף כסידור של קודקודיו לאורך קו אופקי, כך שכל הקשתות המכוונות פונות משמאל לימין.

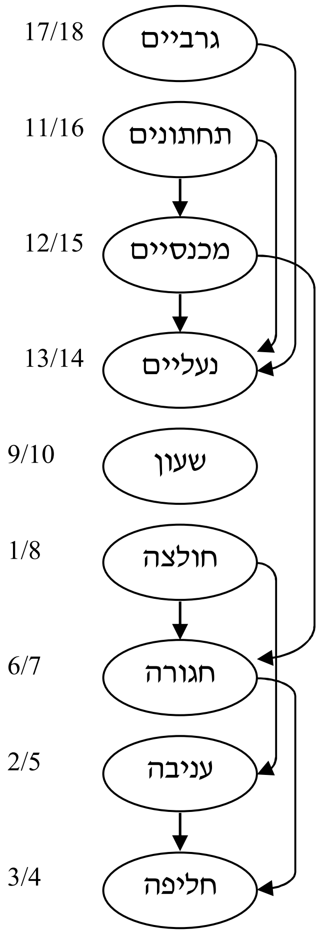
לדוגמא: נתון הגרף המכוון ללא מעגלים (גמ"ל) שלעיל, ומתחתיו מתואר הגמ"ל הזה לאחר מיון טופולוגי. נשים לב שלו הגרף היה מכיל מעגלים, אז לא ניתן לסדר את קודקודיו בסדר לינארי, ולכן לא היה קיים לו מיון טופולוגי. זו הסיבה מדוע אנחנו דורשים שהגרף G יהיה גמ"ל.

בגרפים מכוונים ללא מעגלים ניתן להשתמש ביישומים שונים, לדוגמא – כדי לציין קדימויות בתוך קבוצת מאורעות. נבנה גרף שבו כל קודקוד ייצג פעולה מסוימת, ותהיה קשת מכוונת בין קודקוד u לקודקוד v, אם חייבים לבצע את הפעולה u לפני שנוכל לבצע את הפעולה v.

לדוגמא: נבנה גרף שבו כל קודקוד ייצג לבישת פריט לבוש מסוים (לדוגמא: לבישת חולצה, או ענידת שעון, וכו'). חייבים ללבוש פריטי לבוש מסוימים לפני אחרים (גרביים לפני נעליים, למשל), ונייצג זאת על-ידי קשת מכוונת בין פריט הלבוש אותו צריך ללבוש קודם, לבין פריט הלבוש שאותו ניתן ללבוש אחריו.



כך יראה אותו הגרף לאחר מיון טופולוגי:



איך קיבלנו את התוצאה? על-ידי האלגוריתם הבא, המקבל כקלט גרף מכוון ללא מעגלים G, ומחזיר רשימה מקושרת המכילה סידור לינארי של קודקודיו:

צעד 1: אתחל רשימה מקושרת ריקה.

צעד 2: הפעל את האלגוריתם DFS על הגרף G, וחשב את מועד הסיום finished עבור כל קודקוד.

צעד 3: כאשר קודקוד מסוים נצבע בשחור ונקבע עבורו מועד סיום (כלומר: כאשר הטיפול בו מסתיים), הכנס אותו לראש הרשימה המקושרת.

צעד 4: החזר את הרשימה המקושרת.

הרשימה המקושרת המוחזרת בצעד מס' 4 מכילה את קודקודי הגרף, כשהם מסודרים בסדר יורד לפי מועדי הסיום (finished) שלהם. מכיוון שהכנסת קודקוד לראש רשימה מקושרת מתבצעת בזמן $\Theta(1)$, והרצת DFS אורכת $\Theta(V+E)$, נקבל כי סיבוכיות זמן הריצה של האלגוריתם למיון טופולוגי היא $\Theta(V+E)$.

<u>שאלה</u>

דרך אחרת לבצע מיון טופולוגי של גרף מכוון ללא מעגלים $G = (V,E)$ היא למצוא פעם אחר פעם קודקוד בעל דרגת כניסה 0, להוציאו לפלט, ולהסיר אותו ואת כל הקשתות היוצאות ממנו מן הגרף. הפעילו אלגוריתם זה על הגרף מהדוגמא, ובדקו שאכן מתקבל מיון טופולוגי. מה יקרה לאלגוריתם זה אם G מכיל מעגלים?

<u>שאלה</u>

הפעילו את האלגוריתם למציאת מיון טופולוגי על הגרף המכוון הבא :