# אלגוריתמים קומבינטוריים
# סיכומים של תרגילי כיתה מסמסטרים קודמים בנושא
# עץ פורש מינימום

1. הגדרות: גרף $G(V,E)$ מוגדרת על ידי קבוצת צמתים (קודקודים, vertices, nodes) $V$ וקבוצת קשתות $E$. כל קשת מחבר זוג צמתים. בגרף מכוון, יש לקשת $(u,v)$ כיוון מהזנב (מקור) שלו ($u$) אל הראש (יעד) שלו ($v$). בגרף לא מכוון לקשת $\{u,v\}$ אין כיוון. (שים לב לשוני בסימון! אך לפעמים לא מבדילים בסימון.) לולאה עצמית היא קשת שמחברת צומת לעצמה. הדרגה של צומת $v$ היא מספר הקשתות הנוגעות בו (כאשר לולאה עצמית נספרת פעמים) ומסומן $\deg(v)$ (לפעמים $\mathrm{d}(v)$).

2. משפט: $\sum_{v\in V}\deg(v) = 2|E|$. (כל קשת נספר פעם אחד בקצה הראשון, ופעם בקצה השני. שים לב לחשיבות ספירת הלולאות פעמים!)

3. הגדרה: מסלול בגרף הוא סדרה מהסוג $v_0, e_1, v_1, e_2, , v_2, \ldots, e_k, v_k$ שבו הקשת $e_i$ מחבר בין $v_{i-1}$ ו- $v_i$. במסלול מכוון הקשת $e_i$ מחבר מ- $v_{i-1}$ ל- $v_i$. מסלול הוא פשוט אם אין בו צומת שמופיע פעמיים. מעגל הוא מסלול שבו $v_k = v_0$ ופרט לזה אין צומת שמופיע פעמיים.

4. הגדרה: גרף הוא קשיר אם לכל זוג של קודקודים יש מסלול המחבר אותם.

5. הגדרה: יער הוא גרף חסר מעגלים. גרף קשיר חסר מעגלים נקרא עץ.

6. בהנתן גרף קשיר $G(V,E)$, עץ פורש הוא תת-גרף $T(V,E')$ על $V$ שהוא קשיר וחסר מעגלים. (הוא נקרא פורש כי חוא מקשר את כל הצמתים.)

7. בעיה: בהנתן גף קשיר $G(V,E)$ ופונקצית משקל $w: E \to \mathbf{R}^+$ מצא עץ פורש במשקל מינימום. משקל העץ $T(V,E')$ הוא $w(T) = \sum_{e\in E'} w(e)$.

8. ניתן לפתור את הבעיה על ידי שימוש באלגוריתם חמדני. (עיין בהרצאות.)

9. יש שני אלגוריתמים מפורסמים לבעיה (כל אחד חמדני במובן מסוים). הראשון הוא האלגוריתם של קרוסקל (Kruskal) השני הוא האלגוריתם של פרים (Prim).

10. באלגוריתם של קרוסקל מוסיפים צלעות בופן חמדני (כך שלא ייצר מעגל) עד שמקבלים עץ. במשך האלגוריתם תמיד יש יער. הרעיון: אתחל את $E' = \emptyset$. מיין את הקשתות לפי משקלם בסדר לא-יורד. עבור על הקשתות בסדר זו: כאשר מטפלים בקשת $e$ הוסף אותו ל- $E'$ אלא אם כן הוא יסגור מעגל בגרף $G'(V,E')$ הנוכחי. ניתן לעצור את התהליך כאשר $|E'| = |V| - 1$.

11. מימוש אחד: (תבדקו בתרגילי בית שזה אכן עובד!)

```
KRUSKAL(G, w)
1   E' ← ∅
2   for v ← 1 to |V|
3       do c[v] ← v
4   NON-DEC-SORT(E, w)    ▷ Sort E in nondecreasing order by weight.
5   for i ← 1 to |E|
6       e ← E[i]
7       {u, v} = e    ▷ Defines u and v. Assume that u < v.
8       if c[u] ≠ c[v] then
9        do E' ← E' ∪ {e}
10           x ← c[u]; y ← c[v]
11           for w ← 1 to |V|
12            do if c[w] = y then c[w] ← x
13  return (E')
```

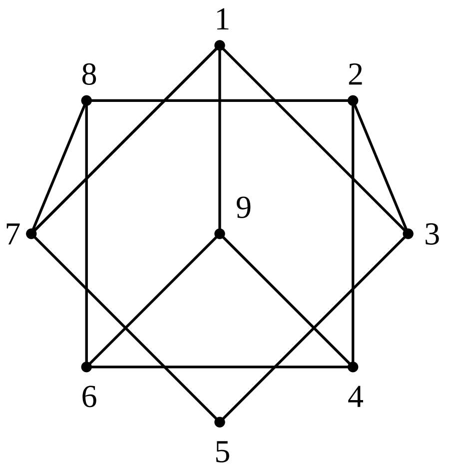
מימוש זה רצה בזמן $O(|V|^2 + |E|\log|E|)$ כי האיתחול דורש זמן $O(|V|)$, המיון זמן $O(|E|\log|E|)$, וסריקת הקשתות $O(|V|^2)$ בגלל התיקונים ל- $c$ בשורות 11-12 בכל אחד מה- $|V|-1$ פעמים שמוסיפים קשת ל- $E'$. מימוש מתוכחם יותר רצה בזמן $O(|E|\log|E|)$.

12. באלגוריתם של פרים מגדלים עץ על תת-קבוצה של הצמתים כאשר שאר הצמתים נשארים בודדים. בכל צעד מוסיפים את הקשת הזולה ביותר מהעץ הקיים לצומת מחוץ לעץ. מימוש פשוט דורש זמן $O(|V||E|)$ אך ניתן לממש את האלגוריתם בזמן ריצה $O(|E|+|V|\log|V|)$.

13. קוד בסיסי:

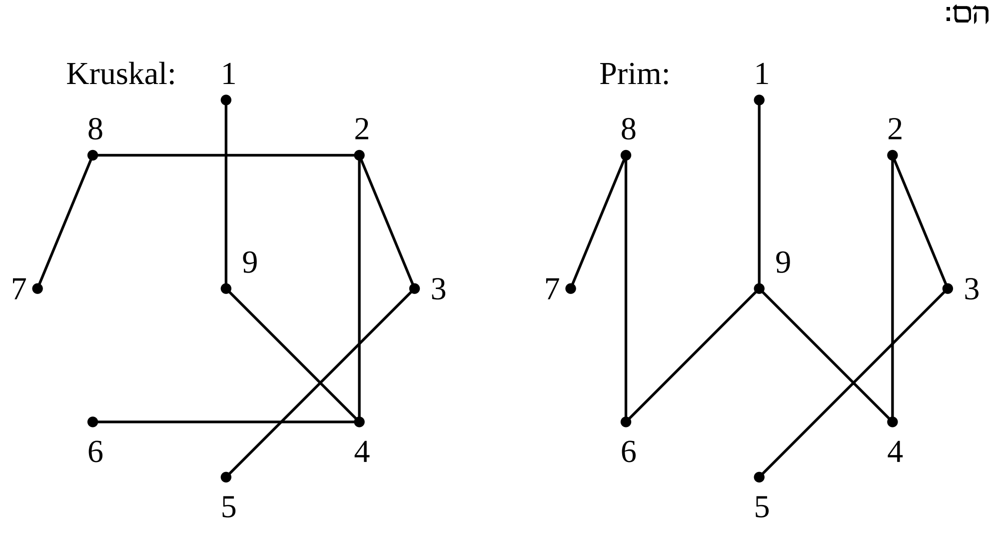
$\mathrm{PRIM}(G, w)$
$E' \leftarrow \emptyset$
$U \leftarrow \{1\}$ $\quad \triangleright$ Vertices numbered $1, 2, \ldots, |V|$.
**while** $U \neq V$
  **do** Find a minimum weight edge $(u, v)$ from $U$ to $V \setminus U$
    $U \leftarrow U \cup \{v\}$
    $E' \leftarrow E' \cup \{e\}$
**return** $(E')$

14. דוגמה: נתון הגרף הבא אם המשקלות בטבלה הרשומה.




| $e$ | $(1,3)$ | $(1,7)$ | $(1,9)$ | $(2,3)$ | $(2,4)$ | $(2,8)$ | $(3,5)$ | $(4,6)$ | $(4,9)$ | $(5,7)$ | $(6,8)$ | $(6,9)$ | $(7,8)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $w$ | 7 | 7 | 3 | 5 | 4 | 4 | 6 | 3 | 3 | 6 | 4 | 3 | 5 |

באלגוריתם של קרוסקל אנחנו נשאיר קשתות במשקל שווה בסדר שרשומים בטבלה (משמאל לימין). באלגוריתם של פרים במקרה שיש שתי קשתות זולות ביותר, אנחנו נבחור קשת היוצאת מצומת שנוספה לעץ החלקי מוקדם יותר ומבין אלו לצומת שמספרו קטנה יותר. כללים אלו שוברים מצבי "תיקו" באשר לבחירת קשת הבא לטיפול באלגוריתמים. אם נתנהג לפיהם, נקבל את העצים נמצוירים למטה. (הדגם את ריצות האלגוריתמים ובדוק!!!) שתי האלגוריתמים מספקים עצים פורשים במשקל 33 ואלו הם:




שים לב שקבלנו שני עצים שונים! מצב זו ייתכן בגלל שיש קשתות שונות בעלי אותה משקל!