הרצאה 4: אלגוריתמים חמדניים

מרצה: אורי שטמר

Textbook: Cormen, Leiserson, Rivest, Stein. Introduction to Algorithms.

בעיית חלוקת הפעילויות

<u>תאור לא פורמלי של הבעיה:</u> הקלט הוא אוסף של n פעילויות. המטרה היא לשבץ את הפעילויות במספר קטן ככל האפשר של חדרים, כך שלא יהיו פעילויות חופפות באותו חדר.

<u>בצורה פורמלית:</u>

<u>הקלט:</u> אוסף של n פעילויות. לכל פעילות i יש זמן התחלה s_i וזמן סיום f_i כך ש- $s_i < f_i$

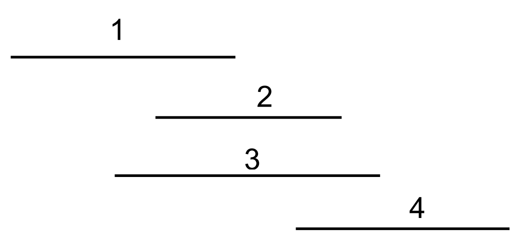
<u>פתרון חוקי:</u> חלוקה של $\{1,2,\dots,n\}$ למספר קבוצות $G_1, G_2, \dots, G_d$ כך שהפעילויות בכל G_i זרות זו לזו.

תזכורת להגדרת חלוקה:
1. $G_i \cap G_j = \emptyset$ לכל $i \neq j$
2. לכל $\ell \in \{1,2,\dots,n\}$ קיים i כך ש- $\ell \in G_i$

<u>יש למצוא:</u> פתרון חוקי עם מספר קטן ככל האפשר של קבוצות.

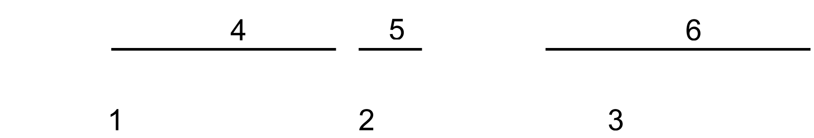
<u>רעיון לאלגוריתם חמדן שלא עובד:</u>
* שבץ מספר גדול ככל האפשר של פעילויות לחדר הראשון (עפ"י אלגוריתם קודם)
* מבין הפעילויות שנשארו, שבץ מספר גדול ככל האפשר לחדר השני
* וכן הלאה, עד שנכסה את כל הפעילויות

<u>דוגמה שבה זה כן עובד:</u>



$\{1,4\}$
$\{2\}$
$\{3\}$

<u>דוגמה נגדית:</u>



הפתרון האופטימלי הוא $\{4,5,6\}$, $\{1,2,3\}$
אבל האלגוריתם שהצענו יתחיל עם $\{1,5,3\}$ ואז $\{4,6\}$ ואז $\{2\}$

<u>אלגוריתם חמדן שעובד:</u>

(1) $d \leftarrow 0$ (מספר החדרים בפתרון שאנחנו בונים)

(2) מיין את הפעילויות עפ"י זמן התחלה. נקבל מיון $i_1, i_2, \dots, i_n$ כך ש- $s_{i_1} \le s_{i_2} \le \dots \le s_{i_n}$.

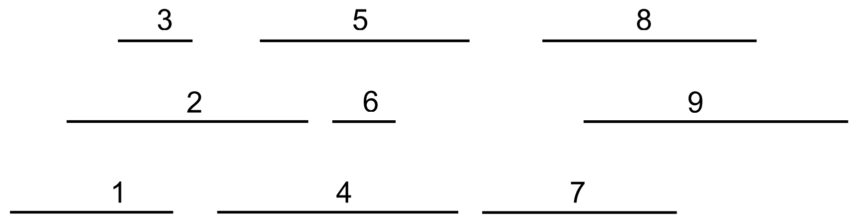
(3) עבור $k = 1$ עד n בצע: (עוברים על הפעילויות עפ"י סדר המיון)

– אם אפשר להכניס את הפעילות i_k לאחד החדרים הקיימים אז הכנס אותה.

– אחרת

$$d \leftarrow d + 1$$
$$G_d \leftarrow \{i_k\}$$

<u>דוגמת ריצה:</u> (נניח שהפעילויות כבר ממוינות עפ"י סדר זמן התחלה)



בהתחלה $d = 0$ כלומר אין חדרים ולכן לא ניתן להכניס את הפעילות 1 לאף חדר. לכן נקבל $d = 1$ ו- $G_1 = \{1\}$

אח"כ א הפעילות 2 אי אפשר להכניס לחדר הקיים (כי היא מתנגשת עם פעילות 1) ולכן $d = 2$ ו- $G_2 = \{2\}$

כך גם עבור פעילות 3 ונקבל $d = 3$ ו- $G_3 = \{3\}$

אח"כ את פעילות 4 אנחנו יכולים להכניס גם לחדר G_1 וגם לחדר G_3. נכניס אותה למשל ל- G_1 ונקבל $G_1 = \{1,4\}$

אח"כ נכניס את פעילות 5 ל- G_3, כלומר $G_3 = \{3,5\}$

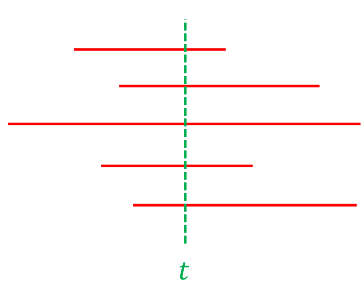
אח"כ נכניס את פעילות 6 ל- G_2, כלומר $G_2 = \{2,6\}$

וכן הלאה...

בקיצור: כל פעם מוסיפים איבר לחדר קיים אם אנחנו יכולים ואחרת נפתח עוד חדר.

<u>הוכחת נכונות:</u>

<u>אבחנה:</u> אם קיים זמן t ו- d פעילויות שמתבצעות בזמן t אזי בכל פתרון חוקי יהיו לפחות d חדרים.

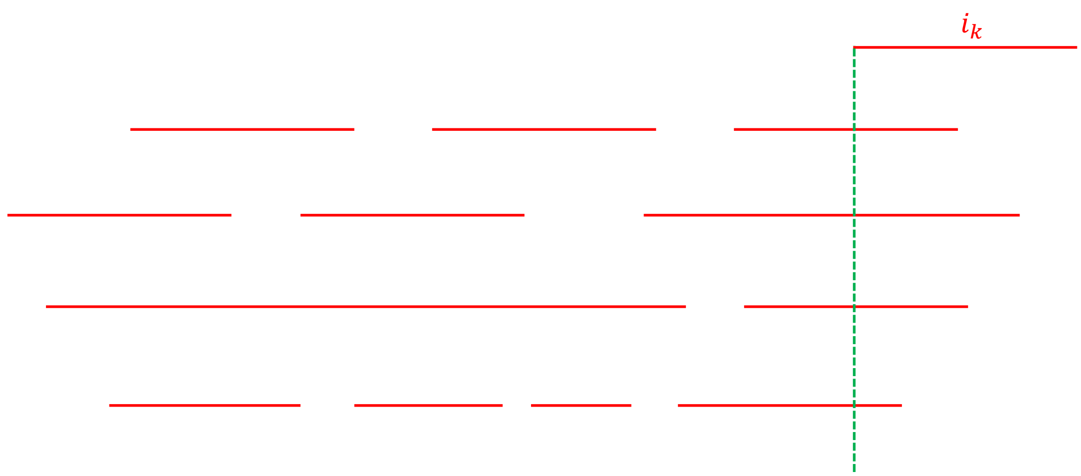


בציור: אם יש פחות מ- d חדרים אז 2 פעילויות כאלה משוצות לאותו חדר אבל אז הן מתנגשות...

טענה: אם בשלב ה- k של האלגוריתם פתחנו חדר חדש, ז"א הגדלנו את d ל- $(d+1)$, אזי ישנם $(d+1)$ פעילויות שמתבצעות בו זמנית.

הוכחת הטענה:
נסתכל על הזמן s_{i_k} = זמן ההתחלה ה- k בסידור שעליו אנו מסתכלים בשלב ה- k.
בכל קבוצה $G_1, \ldots, G_d$ קיימת פעילות שנחתכת עם פעילות k.

לדוגמה:



עפ"י האלגוריתם, פעילויות אלו התחילו לפני s_{i_k} ועדיין לא הסתיימו.
כלומר בכל חדר קיימת פעילות שהתחילה לפני s_{i_k} ומסתיימת אחרי s_{i_k} ובפרט מתבצעת בזמן s_{i_k}.
כלומר, קיימות $d+1$ פעילויות שמתבצעות בזמן s_{i_k}.
מ.ש.ל.

הערה: בהגדרת הבעיה הנחנו שהפעילויות שלנו לא כוללות את קצוות אינטרוולי הזמן שלהם, כלומר הפעילות i_k מתבצעת בזמן $s_{i_k} < t < f_{i_k}$ (ולא בזמן $s_{i_k} \le t \le f_{i_k}$). לכן טכנית הפעילות i_k לא מתבצעת בזמן s_{i_k}. פורמלית, כדי להתחשב בזה בהוכחה, היינו צריכים לדבר על זמן $s_{i_k} + \varepsilon$ עבור ε מספיק קטן.

משפט: האלגוריתם מחזיר פתרון חוקי עם מספר מנימלי של חדרים.

הוכחה:

- האלגוריתם מחזיר פתרון חוקי:
  - האלגוריתם עובר על כל הפעילויות ומשבץ כל פעילות לבדיוק חדר אחד (כלומר הפתרון שהאלגוריתם מחזיר הוא <u>חלוקה</u> של הפעילויות לחדרים)
  - אם האלגוריתם מכניס פעילות לחדר אז היא זרה לכל הפעילויות שבחדר.

- הפתרון המוחזר אופטימלי:
  - נסתכל על ה k האחרון שפותח חדר, חדר מספר $\tilde{d}$. עפ"י הטענה, ישנם $\tilde{d}$ משימות בו זמנית
  - עפ"י האבחנה, כל פתרון משתמש בלפחות $\tilde{d}$ חדרים
  - הפתרון שמוחזר ע"י האלגוריתם משתמש ב $\tilde{d}$ חדרים ולכן אופטימלי

<u>מימוש וניתוח זמן ריצה:</u>

* המיון מתבצע בזמן $O(n \cdot \log n)$
* מימוש הלולאה: נחזיק לכל קבוצה G_j את זמן סיום הפעילות האחרונה בקבוצה, שנסמנו כ- F_j. כדי לדעת אם אפשר להכניס את i_k לחדר G_j נבדוק אם $F_j \leq s_{i_k}$

במקרה הגרוע יש $O(n)$ קבוצות ולכן זמן הריצה הכולל יהיה $O(n^2)$, כי עבור כל פעילות נבדוק $O(n)$ חדרים.

איך ניתן לממש את האלגוריתם בצורה יעילה יותר?

נשים לב: במימוש הנ"ל, כשאנחנו בודקים האם קיים חדר אליו i_k יכולה להיכנס, או שאנחנו צריכים לפתוח חדר חדש, אנחנו בדקנו לכל חדר G_j האם $F_j \leq s_{i_k}$.

<u>מימוש יעיל יותר:</u>

* נסתכל על החדר עם זמן סיום <u>מינימלי</u> ונבדוק אם אפשר להכניס את i_k לחדר הזה.
* נחזיק את $F_1, F_2, \ldots, F_d$ בערימה.
  - מציאת חדר עם זמן סיום מינימלי (find_min) מתבצע בזמן $O(1)$
  - אחרי הכנסת איבר ל- G_j צריך להוציא את F_j מהערימה (del_min) ולהכניס זמן סיום חדש (insert). כל אחת משתי הפעולות האלה מתבצעת בזמן $O(\log n)$
* ישנן n פעולות כנ"ל ולכן סה"כ זמן ריצה $O(n \cdot \log n)$

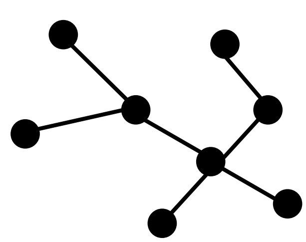
בעיית עץ פורש מינימום (עפ"מ)

בהרצאה הבאה נדבר על בעיית "עץ פורש מינימום". היום רק נזכר מהו בעצים/עצים פורשים ובכמה תכונות שלהם.

שימו לב: באתר הקורס התפרסם קובץ עם תזכורות למושגים מתורת הגרפים שלמדתם בקורסים קודמים. יתכנו הבדלים קטנים בין מונחים והגדרות בתכנון אלגוריתמים לעומת קורסים קודמים. אתם מתבקשים לעבור על הקובץ הזה.

<u>הגדרה:</u> <u>עץ לא מכוון</u> הוא גרף קשיר חסר מעגלים

לדוגמה



<u>משפט:</u> ארבעת התנאים שקולים עבור גרף לא מכוון $T = (V, E)$

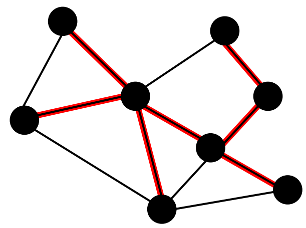
1. T הוא עץ, כלומר גרף קשיר חסר מעגלים
2. T קשיר ומכיל $|V| - 1$ קשתות
3. T חסר מעגלים ומכיל $|V| - 1$ קשתות
4. בין כל זוג צמתים ב- V יש בדיוק מסלול פשוט אחד

(המשפט הזה יהיה חשוב לנו כשנרצה להוכיח שהאלגוריתם שלנו מחזיר עץ. אז במקום להראות שהגרף המוחזר הוא קשיר וחסר מעגלים, לפעמים יהיה לנו נח להראות תנאי אחר מהרשימה)

<u>הגדרה:</u> בהינתן גרף לא מכוון $G = (V, E)$, הגרף $T = (V', E')$ הוא <u>עץ פורש</u> של G אם:

1. $V' = V$
2. $E' \subseteq E$
3. T הוא עץ

דוגמה:



<u>טענה:</u> לכל גרף קשיר ולא מכוון $G = (V, E)$ קיים עץ פורש

<u>רעיון ההוכחה:</u> באינדוקציה על $|E|$

<u>בסיס:</u> $|E| = 0,1$ טריוויאלי

<u>צעד:</u> נניח שלכל גרף עם לכל היותר $m - 1$ קשתות יש עץ פורש ויהי $G = (V, E)$ גרף עם m קשתות.
תהי $e \in E$ קשת כלשהי.
אם $G' = (V, E \setminus \{e\})$ קשיר אז יש לו עץ פורש לפי הנחת האינדוקציה (וזהו גם עץ פורש של G).
אחרת יהיו G_1, G_2 שני הגרפים הקשירים המתקבלים כתוצאה ממחיקת e מ- G.
לפי הנחת האינדוקציה לכל אחד מהם יש עץ פורש $T_1 = (V_1, E_1),\ T_2 = (V_2, E_2)$ בהתאמה.
נתבונן בגרף $T = (V, E_T)$ כאשר $E_T = E_1 \cup E_2 \cup \{e\}$.
ניתן לוודא ש- T הוא קשיר. בנוסף, מספר הקשתות בו הוא
$$|E_T| = |E_1| + |E_2| + 1 = (|V_1| - 1) + (|V_2| - 1) + 1 = |V_1| + |V_2| - 1 = |V| - 1$$
ולכן הוא עץ.