פרטים אדמיניסטרטיביים

- מרצה: עדן כלמטץ'
- שעות קבלה:
  - יום א' 16:00-18:00
  - בניין 37, חדר 210
- מייל: chlamtac@cs.bgu.ac.il
- אתר הקורס: http://www.cs.bgu.ac.il/~algo142
- מייל לפניות: algo142@cs.bgu.ac.il
- תהיינה 6 עבודות בית
  - לוקחים את הציונים של 5 העבודות הטובות ביותר
  - ההגשה היא בזוגות – אפשר להגיש לבד, אבל עדיף שלא
- יהיה בוחן **אופציונאלי** (כלומר ניתן לבחור אם לגשת או לא)
- מי שיעתיק – האוניברסיטה תכסח אותו
- ישנה חובת נוכחות בהרצאה הראשונה של כל שבוע, שכן התרגול באותו שבוע מבוסס עליה
- הקורס הזה אינו קורס קל [[אבל אני מאמין בכם! תראו להם מה זה! ☺]]

הקדמה

היה היה מתמטיקאי בשם פאול אֶרדֶש (Erdős Pál), שחי בשנים 1913-1996.

הוא פרסם **מעל ל-1500 מאמרים**!
הצלחתו הרבה נבעה מכמה דברים:

- כישרון
- שימוש נרחב בסמים
- שיתוף פעולה מדעי עם אנשים אחרים (סה"כ 458 מחברים פרסמו איתו מאמר)

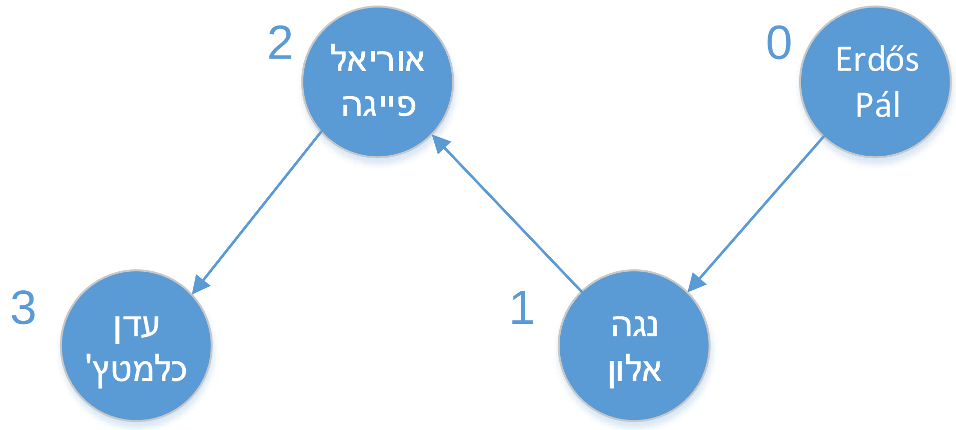
על שמו יש מושג הנקרא מספר ארדש.

הגדרה [מספר ארדש]

נגדיר גרף, שקדקודיו הם מחברים, והקשתות בו מציינות ששני המחברים עליהם הקשת חלה פרסמו יחד מאמר.
לכל מחבר נגדיר מספר:

- המספר של ארדש עצמו הוא 0.
- המספר של כל שכניו הישירים הוא 1,
- המספר של כל השכנים הישירים שלהם הוא 2
- וכן הלאה...

כלומר, מספר ארדש של מחבר מסוים שווה למרחק בין אותו מחבר לבין ארדש בגרף המתואר.



שאלת דוגמה

נשאל את השאלה: כיצד, בהינתן הגרף הזה, ניתן לדעת מהו מספר ארדש של מחבר מסוים?
כלומר, איזה אלגוריתם ניתן להפעיל עליו כדי לקבל את המידע הזה?

התשובה לשאלה זו היא אלגוריתם BFS, שלמדנו בקורס *מבני נתונים* (מס' 202-1-1031).

תכנון אלגוריתמים

לכל בעיה:

- נמצא פתרון נאיבי
- נתכנן אלגוריתם בשיטת Top-down:
  - נתאר את השלבים העיקריים של האלגוריתם
  - נוכיח נכונות:
    - ננסח משפט מרכזי [בגדול הוא לרוב יאמר ש"האלגוריתם עובד"]
    - ננסח טענות עזר
    - נוכיח את המשפט על-סמך טענות העזר
    - נוכיח את טענות העזר
  - נכתוב פרטי מימוש
  - ננתח זמן ריצה

[[הערה: הסיבה שקודם מוכיחים את המשפט בעזרת טענות העזר ורק אז מוכיחים את טענות העזר היא שיתכן שנבין במהלך הוכחת המשפט שאנו צריכים טענות עזר שונות, אז חבל להשקיע זמן ומאמץ בלהוכיח משהו שבסוף לא נצטרך.]]

הגדרת בעיה

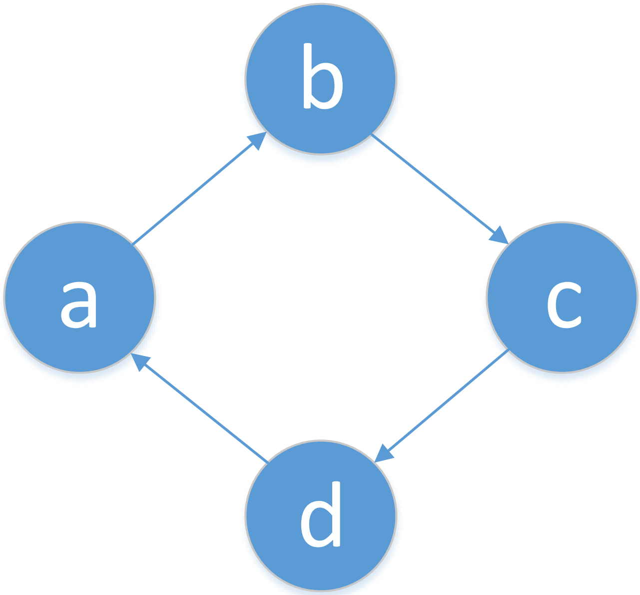
בעיה מתארת קשר בין קלט לפלט.
נגדיר אותה ע"י:

- **מופע**: קלט חוקי לבעיה.
- יש למצוא: פלט חוקי (כלשהו), אם קיים כזה.

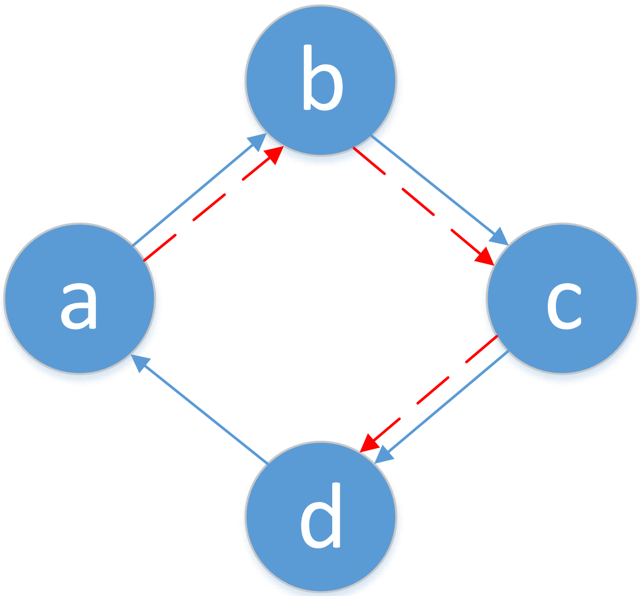
דוגמה [מסלול המילטון]

- מופע: גרף מכוון $G=(V,E)$.
- יש למצוא: מסלול (מכוון) שמתחיל ומסתיים בקדקודים שונים ועובר בכל קדקוד בדיוק פעם אחת.

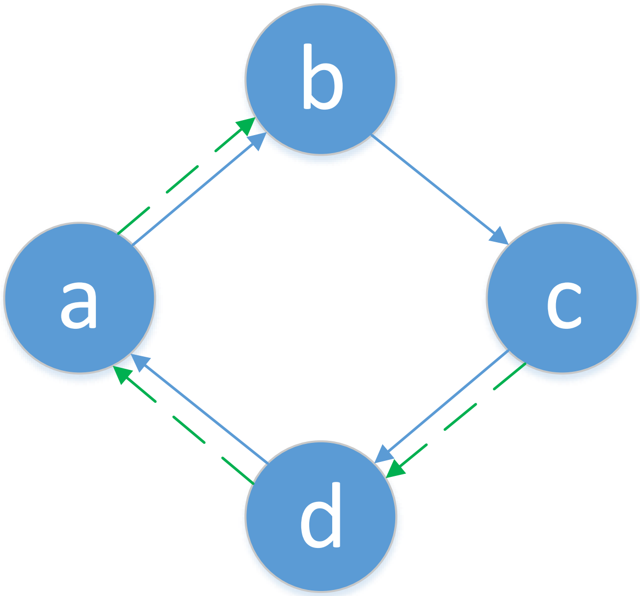
דוגמאות לקלטים ופלטים



פלטים חוקיים:

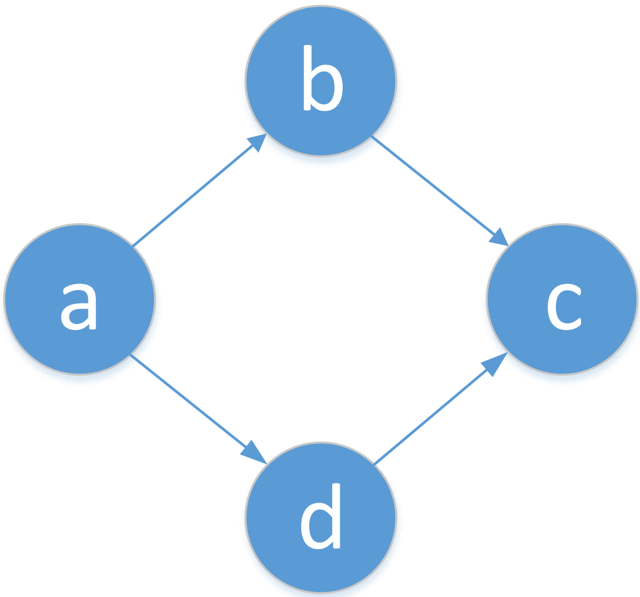


$\langle a,b,c,d \rangle$



$$\langle c, d, a, b \rangle$$

<u>עוד דוגמה</u>:



במקרה זה אין פתרון.

פתרון נאיבי

- לעבור על כל הפרמוטציות של הקדקודים
- לבדוק עבור כל אחת האם היא מסלול תקין [הרי יתכן שאין אחת מהקשתות שמופיעות במסלול]

<u>זמן ריצה:</u>

כמספר הפרמוטציות:

$$O(|V|!) = O\left(2^{\log(|V|!)}\right) \overset{\substack{\log(n!)=O(n\log n)\\ \downarrow}}{=} O\left(2^{|V|\log|V|}\right) \gg 2^{|V|}$$

אלגוריתם זה מאוד לא יעיל.

רדוקציות

רדוקציה מבעיה A לבעיה B היא אלגוריתם שפותר את A באמצעות B.

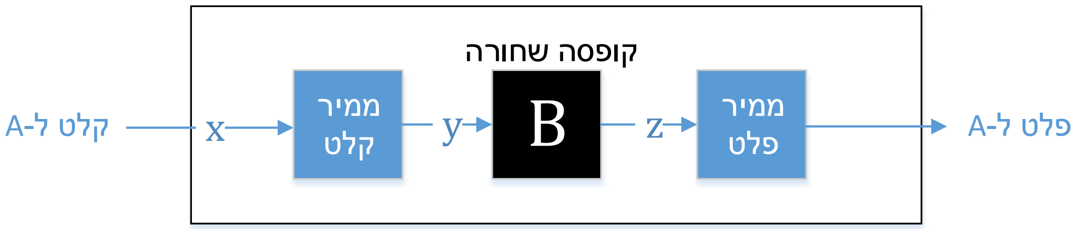
[[כלומר, יש לנו "קופסה שחורה" (זה שם למשהו שאנו יודעים מה הוא עושה, אך לא מתעניינים **באיך** הוא עובד), שיודעת עבור קלט ל- B להוציא פלט נכון. בשביל לפתור את A, אנו ממירים את הקלט ל- A לקלט שמתאים ל- B, מפעילים על הקלט המומר את הקופסה השחורה שפותרת את B ואז ממירים את התוצאה לתוצאה שמתאימה ל- A.]]

לדוגמה, נניח שאנו יודעים לפתור את הבעיה "מהו סכום של שני מספרים בינאריים?"
כלומר יש לנו אלגוריתם שעבור כל קלט שניתן לו, נותן לנו פלט חוקי ונכון עבור בעיה זו.
כעת אנו רוצים לפתור את הבעיה "מהי המכפלה של שני מספרים בינאריים?"
[נראה את זה לעומק בקורס *מבוא למחשבים*.]

בתור קלט אנו מקבלים מספרים $x = x_{n-1}x_{n-2}\cdots x_1x_0$ ו- $y = y_{n-1}y_{n-2}\cdots y_1y_0$.
הפלט יהיה סכום של מספרים מהצורה $z = z_{m-1}z_{m-2}\cdots z_1z_0$, כאשר כל אחד מהם הוא תוצאה של מכפלת x עם מספר מהצורה $000\cdots0y_i0\cdots0$.
[[מצטער, אין לי פירוט ממש כי לא היה לי מקום בכיתה שממנו יכולתי לראות את כל הלוח.]]

רדוקציית קארפ

[[זהו מבנה כללי לרדוקציה, שבו בד"כ משתמשים.]]
תיאור סכמטי:



אלגוריתם מבוסס רדוקציה לבעיה B עבור בעיה A:

- בהינתן x, מפעילים ממיר קלט ומקבלים קלט y ל- B.
- מפעילים אלגוריתם ל- B [[על y]] ומקבלים פלט z.
- ממירים על z ממיר פלט ומקבלים את התשובה (פלט של A).

כיוון שאנו יכולים לפתור את A בעזרת B, ניתן להסיק שאם B בעיה "קלה" אז גם A בעיה "קלה", ואם A בעיה "קשה" אז גם B בעיה "קשה" [[זה דומה למשפט הרדוקציה שראינו בקורס *אוטומטים, שפות פורמליות וחישוביות*]].

דוגמה [מסלול קצר ביותר בגרף לא מכוון בעזרת גרף מכוון]

- בעיה A: מציאת מסלול קצר ביותר מ- s ל- t בגרף **לא מכוון**.
- בעיה B: מציאת מסלול קצר ביותר מ- s ל- t בגרף **מכוון**.

פתרון:

את בעיה B אנו כבר יודעים שניתן לפתור בעזרת BFS.

- ממיר קלט: בגרף הנתון $G=(V,E)$ נחליף כל קשת בשתי קשתות מכוונות בכיוונים שונים ונחזיר גרף G'.
- ממיר פלט: נחזיר את אותו המסלול (אותה סדרת קדקודים).

הוכחה

משפט: האלגוריתם מבוסס הרדוקציה מחזיר מסלול קצר ביותר מ- s ל- t.
טענת עזר: כל סדרת צמתים P היא מסלול באורך l ב- G אם"ם P היא מסלול באורך l ב- G'.

זמן ריצה

- ממיר קלט: שכפול הקשתות – $O(|E|)$.
- אלגוריתם ל-B: BFS מ- s: $O(|V|+|E|)$.
- ממיר פלט: $O(1)$ [הוא לא מבצע כלום].
- סה"כ: $O(|V|+|E|)$.

למה צריך רדוקציות?

- בניית אלגוריתמים (יעילים)
- הוכחות קושי [בדומה למה שעשינו בקורס *אוטומטים, שפות פורמליות וחישוביות*]:
  - "אם אפשר לפתור את B, אז אפשר לפתור גם את A"
  - "אם אי-אפשר לפתור את A, אז אי-אפשר לפתור גם את B"

  [שתי הטענות שקולות, אך לכל אחת מהן יש שימושים משלה.]

בעיית התת-מחרוזות

- מופע: אוסף של n מחרוזות שונות באורך 3:

$$W=\{w_1,w_2,\dots,w_n\}$$

- יש למצוא: מחרוזת T באורך $n+2$ כך שאוסף תת-המחרוזות שלה באורך 3 הוא בדיוק W.

דוגמה:

$$W=\{\text{uri},\ \text{gur},\ \text{ngu},\ \text{ion},\ \text{ben},\ \text{eng},\ \text{rio}\}$$
$$\Downarrow$$
$$T=\text{bengurion}$$

רוצים לפתור את בעיית ההכרעה: האם קיימת T מתאימה? (כן / לא)

אלגוריתם נאיבי

- נעבור על כל הפרמוטציות של W
- לכל אחת, נבדוק האם פרמוטציה זו נותנת מחרוזת חוקית [[בודקים אם זו מחרוזת חוקית בכך שעוברים על כל תתי המחרוזות לפי הסדר שנבחר ובודקים ששתי האותיות האחרונות של כל תת-מחרוזת זהות לשתי האותיות הראשונות של תת-המחרוזת העוקבת]]

זמן ריצה:

$$O(|w|!)=O(n!)$$

רדוקציה מבעיית המחרוזות לבעיית מסלול המילטון

אבחנה: הזוג $\langle w_i, w_j\rangle$ יכול לבוא ברצף כתת-מחרוזת של הפתרון $\Leftrightarrow$ הסיפא באורך 2 של $w_i =$ הרישא באורך 2 של w_j.

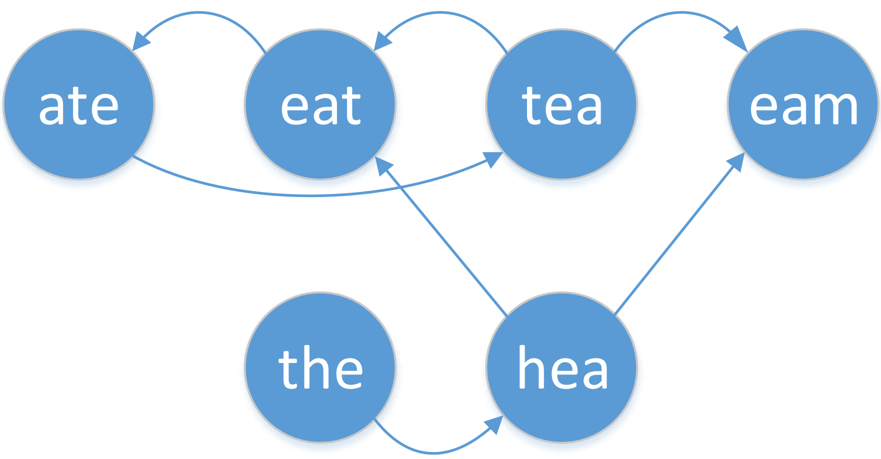
$$\begin{matrix} w_i = & a & b & c & \\ w_j = & & b & c & d \end{matrix}$$

- קלט: W
- ממיר קלט: נבנה גרף $G=(V,E)$ עם צמתים $V=W=\{w_1,\ldots,w_n\}$ וקשת $(w_i,w_j)\in E$ אם"ם הסיפא באורך 2 של $w_i =$ הרישא באורך 2 של w_j.
- ממיר פלט: מקבל פלט מהקופסה השחורה, שהוא או מסלול המילטון, או תשובה "אין מסלול המילטון" [אנו מניחים שהקופסה השחורה נותנת לנו מסלול המילטון, אם יש כזה, או תשובה של "אין מסלול המילטון"], ותחזיר "כן" אם"ם נמצא מסלול המילטון ב- G.

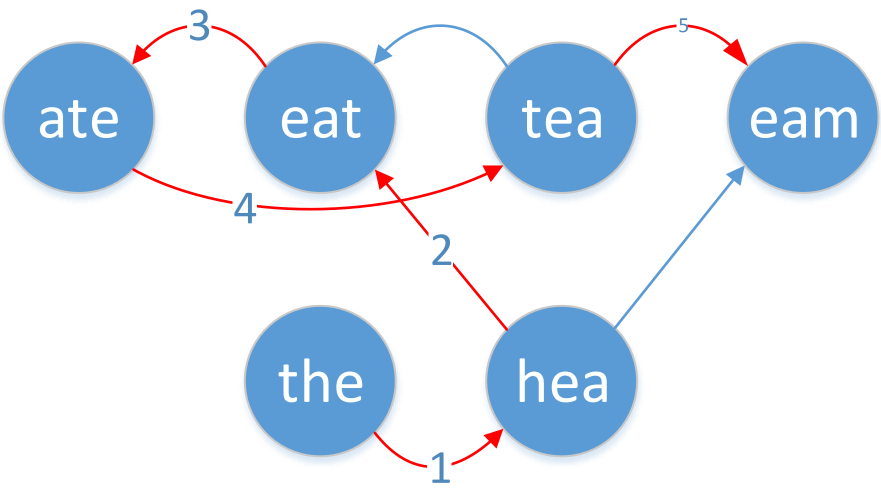
דוגמה:

$$W=\{\text{ate},\ \text{eat},\ \text{tea},\ \text{eam},\ \text{the},\ \text{hea}\}$$

נצייר את G:

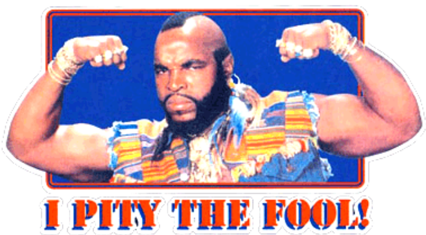


ניתן לראות שיש בגרף מסלול המילטון:



ואז המחרוזת שאנו מקבלים היא:

$$T = \text{theateam}$$



הוכחת נכונות

<u>משפט</u>: האלגוריתם מבוסס הרדוקציה מחזיר "כן" אם"ם קיימת מחרוזת חוקית עבור W.

<u>טענה 1</u>: אם קיימת T חוקית עבור W – אז יש מסלול המילטון ב- G.

<u>טענה 2</u>: אם ב- G יש מסלול המילטון – אזי יש מחרוזת חוקית T עבור W.

[אנו צריכים את שתי הטענות כי עלינו להוכיח שני כיוונים של טענת אם"ם.]

נבצע את ההוכחה בשיעור הבא.

הראינו רדוקציה:

מסלול המילטון $\leq$ בעיית המחרוזות

ואנו מאמינים שהבעיה של מציאת מסלולי המילטון היא בעיה קשה [[נדבר על זה עוד בהמשך הקורס]].