אלגוריתמים 1

״אלגוריתמים + מבני נתונים = תוכנה״

-ניקלאוס וירת

תוכן עניינים

הרצאה 1

תוכן לימודי:

FFT

תכנות דינאמי

אלגוריתמים חמדנים

אלגוריתמים על גרפים

פולינומים

פולינום ממעלה $n-1$ מוגדר כך:

$$A(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$$

כעת ניתן להגדיר עליהם חיבור, כפל ועוד פעולות.

קונבולוציה: פעולה נוספת שממזגת את שני הפולינומים.

<u>שערוך (*evaluation*):</u> בהינתן ערך x_0, נחשב את $y_0 = A(x_0)$.

<u>ייצוג פולינומים:</u> ישנן כמה דרכים לייצג פולינומים.

- ניתן לייצג את הפולינום A על ידי ווקטור $(a_0, a_1, \dots, a_{n-1})$. תצוגה זו מייצגת פולינומים באופן חח״ע.
- ייצוג על ידי קבוצת נקודות: נתונים לנו n ערכים שונים $(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1})$. אזי $A(x)$ יכול להיות מיוצג ע״י הרשימה $(x_0, y_0), \dots, (x_{n-1}, y_{n-1})$.

ניתן להשיג תצוגת מקדמים על ידי n נקודות שונות בצורה הבאה:

$$\begin{pmatrix} x_0^0 & x_0^1 & \cdots & x_0^{n-1} \\ x_1^0 & x_1^1 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-1}^0 & x_{n-1}^1 & \cdots & x_{n-1}^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

כאשר נפתור את המשוואה עבור ווקטור המקדמים.

בעיה: שערוך כל הערכים $y_0, y_1, \dots, y_{n-1}$ לוקח $O(n^2)$.

נראה כיצד ניתן לעשות זאת בזמן $O(n\log(n))$.

שיפור באמצעות כופלי לגראנז':

בהינתן ייצוג על ידי נקודות נרצה לקבל ייצוג על ידי מקדמים.

ניתן להשתמש בכופלי לגראנז':

$$A(x) = \sum y_i \cdot \frac{\Pi_{j\neq i}(x - x_j)}{\Pi_{j\neq i}(x_i - x_j)}$$

פעולות על פולינומים בייצוג נקודות:

בהינתן פולינום $A = (x_0, A(x_0)), (x_1, A(x_1)), \dots, (x_{n-1}, A(x_{n-1}))$

ופולינום נוסף: $B = (x_0, B(x_0)), (x_1, B(x_1)), \dots, (x_{n-1}, B(x_{n-1}))$

נוכל לבצע חיבור בקלות:

$$A + B = (x_0, A(x_0) + B(x_0)), (x_1, A(x_1) + B(x_1)), \dots, (x_{n-1}, A(x_{n-1}) + B(x_{n-1}))$$

עבור כפל, נניח שנתונות לנו $2n - 1$ נקודות שונות המייצגות את כל אחד מהפולינומים. נזדקק ל $2n - 1$ נקודות שונות על מנת לייצג את $A \cdot B$. נקבל:

$$A \cdot B = (x_0, A(x_0) \cdot B(x_0)), (x_1, A(x_1) \cdot B(x_1)), \dots, (x_{2n-1}, A(x_{2n-1}) \cdot B(x_{2n-1}))$$

הערה: אם יש נקודות שונות שמייצגות את A, B, ניאלץ למצוא את המקדמים של שניהם ולמצא נקודות משותפות ורק אז למצוא את המכפלה. זמן ריצה $O(n^2)$.

בניית אלגוריתם ה-FFT

נתונים לנו פולינומים A, B על ידי מקדמים. נרצה למצוא את הכפל בינהם.

רעיון: נתחיל עם ייצוג מקדמים, נעבור לייצוג נקודות, ואז ייצוג נקודות אחרות, ומשם חזרה למקדמים.

נבחר ערכי x שהם שורשי יחידה מסדר n (בעצם, מספרים מרוכבים).

צעד ראשון: שערוך

נציג אלגוריתם DAC שבהינתן פולינום A ממעלה n, ונניח ש-n הוא חזקה שלמה של 2, $n = 2^k$. נרצה לשערך את $A(x)$ ב-n ערכים שונים שהם: $x_0, x_1, \dots, x_{n-1}$.

נפצל את A לחלקים זוגיים ואי-זוגיים:

$$A_{even} = \sum_{i=0}^{\frac{n}{2}-1} a_{2i}x^i, \quad A_{odd} = \sum_{i=0}^{\frac{n}{2}-1} a_{2i+1}x^i$$

כעת נשים לב כי: $A(x) = A_{even}(x^2) + xA_{odd}(x^2)$. כלומר אנחנו אכן יכולים להשתמש בשיטת הפרד ומשול על מנת לפתור את בעיית השערוך.

נחשב כעת את A ב-n ערכים שונים. נבצע זאת באופן רקורסיבי.

תכונת הנגדיות החלשה

יהי k מספר שלם שהוא חזקה שלמה של 2. אזי הרשימה $(x_0, x_1, \dots, x_n)$ היא בעלת תכונת נגדיות חלשה אז או ש-$k = 1$, או שלכל מספרים $n \le r \le \frac{k}{2} - 1$ מתקיים: $x_{\frac{r}{2}} + r = -x_r$

כעת נניח ש-A פולינום ממעלה n, ונתונים לנו n ערכי x שונים שמקיימים את תכונת הנגדיות החלשה:

$$\forall n \le r \le \frac{n}{2} - 1: \left(x_{\frac{n}{2}} + r\right)^2 = (-x_r)^2 + x_r^2$$

ואז סדרת הערכים שלנו (בקריאה רקורסיבית) תהיה:

$$(x_0)^2, (x_1)^2, \dots, \left(x_{\frac{n}{2}-1}\right)^2$$

מדוע אנחנו לוקחים דווקא שורשי יחידה? מכיוון שניתן להגיע בקלות משורשי n לשורשי $\frac{n}{2}$ ולהיפך.

$$w = e^{i\cdot\frac{2\pi}{n}k} \Rightarrow w^{\frac{n}{2}} = e^{i\pi k} = \pm 1$$

<u>למה</u>: לכל $0 \le k \le n-1$, w_n^k הוא שורש יחידה מסדר n.

<u>למה</u>: לכל $n > 1$ חזקה שלמה של 2, ועבור אינדקס $0 \le k \le \frac{n}{2} - 1$, מתקיים:

$$w_n^{k+\frac{n}{2}} = -w_n^k$$

<u>למת החצייה</u>: $w_n^0, w_n^1, \dots, w_n^{n-1}$ מקיימים תכונת נגדיות חלשה.

<u>למה</u>: יהי $n > 1$ חזקה שלמה של 2, אזי ריבוע הערכים של $w_n^0, w_n^1, \dots, w_n^{\frac{n}{2}-1}$ הם שורשי היחידה מסדר $\frac{n}{2}$.

<u>למה</u>: n שורשי היחידה מקיימים את תכונת הנגדיות החזקה.

<u>הוכחה</u>: באינדוקציה על n.

בסיס: עבור $n = 1$, טריוויאלי.

עבור $n > 1$, לפי הלמה, ריבוע הערכים הנ"ל הם שורשי היחידה מסדר $\frac{n}{2}$. לפי למה ___ הם חייבים גם לקיים את תכונת הנגדיות החלשה.

הרצאה 2

<u>תזכורת</u>

למת החציה:

$$w_n^0, w_n^1, \dots, w_n^{\frac{n}{2}-1}$$

הם $\frac{n}{2}$ שורשי היחידה מסדר $\frac{n}{2}$.

<u>למה</u>: n שורשי היחידה מסדר n הם בעלי תכונת ניגודיות חזקה.

<u>אבחנה</u>: כל n שורשי היחידה מסדר n הם שונים.

אלגוריתם FFT

נבנה פסאודו-קוד לאלגוריתם:

```
import numpy as np

def FFT(A):
    n = len(A)
    if n == 1:
        return A  # Base case: return the single element

    # Divide: Split even and odd indexed elements
    A_even = A[0::2]  # Elements at even indices
    A_odd = A[1::2]   # Elements at odd indices

    # Recursive calls
    Y_even = FFT(A_even)
    Y_odd = FFT(A_odd)

    # Combine step
    Y = [0] * n
    wn = np.exp(-2j * np.pi / n)  # Primitive nth root of unity
```

אינטרפולציה

תזכורת: המטרה הייתה לחב מכפלה של שני פולינומים. מה שנשאר הוא לתרגם בחזרה פולינום C עם ייצוג ע״י נקודות לפולינום עם ייצוג ע״י מקדמים.

$$FFT = V = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w_n & w_n^2 & \cdots & w_n^{n-1} \\ 1 & (w_n^2) & (w_n^2)^2 & \cdots & (w_n^2)^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & (w_n^{n-1}) & (w_n^{n-1})^2 & \cdots & (w_n^{n-1})^{n-1} \end{bmatrix}$$

האלגוריתם FFT שהצגנו בעצם מחשב את:

$$V \cdot \bar{a} = \bar{y} \Rightarrow * \ \bar{a} = V^{-1}\bar{y}$$

על מנת לחשב אינטרפולציה עלינו לחשב את *.

טענה:

$$V^{-1} = \frac{1}{n}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & (w_n^{-1}) & (w_n^{-1})^2 & \cdots & (w_n^{-1})^{n-1} \\ 1 & (w_n^{-2}) & (w_n^{-2})^2 & \cdots & (w_n^{-2})^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \left(w_n^{-(n-1)}\right) & \left(w_n^{-(n-1)}\right)^2 & \cdots & \left(w_n^{-(n-1)}\right)^{n-1} \end{bmatrix}$$

Inverse FFT

מסקנה: המטריצה V גם היא מטריצת וונדרמונדה, ולכן האלגוריתם שהצגנו יודע לחשב באופן מדיד את $V^{-1}y$, שזה נותן את ווקטור המקדמים של הפולינום C.

תכנות דינאמי

עובדה מעניינת: תכנות דינאמי נקרא כך מכיוון שהממציא של השיטה (ריצ׳רד בלמן) לא רצה שידעו שהוא בפועל מתעסק במתמטיקה, והוא חיפש שם מרשים לנושא, שייתן לו מימון למחקר. (בכיתה ציינו שזה בגלל השמירה של הנתונים בטבלה, אך הסיבה הזאת לא מדוייקת).

> *"I thought, let's take a word that has an absolutely precise and specific meaning—namely, dynamic—and use it in a way that does not have a precise or specific meaning. But it was something that sounded impressive. Dynamic programming was something not even a Congressman could object to."*

מבנה פתירת בעיות דינאמיות:

1. מקרה בסיס – מה המקרה הכי פשוט של הבעיה?
2. נבדוק כיצד ניתן לפרק בעיה כללית לתתי בעיות מאותו הסוג.
3. הוספת טבלה: נוסיף טבלה לשמירת ערכים, כדי להימנע מלחשב את אותם ערכים.
4. מעבר למבנה $Bottom - Up$.

דוגמא: מצא את מספר פיבונאצ׳י ה-n.

```
F[0] = 1
F[1] = 1
For(i=2 to n)
  F[i]=F[i+1]+F[i-2]
```

הגדרה: נאמר שתת מערך של A עם אינדקסים $i_1, i_2, \dots, i_m$ הוא בלתי תלוי, אם לכל k: $i_k < i_{k+1} - 1$.

<u>מטרה</u>: מצא את תת הסדרה הבלתי תלויה עם הערך המקסימלי של A.

<u>פתרון נאיבי</u>: נבדוק את כל תתי הסדרות הב"ת, ונחזיר את הסדרה עם הערך המקסימלי. יש לנו 2^n סדרות שונות, ולוקח זמן של $O(n)$ לחשב את הערך של כל אחד מהן. סך הכל $O(n \cdot 2^n)$.

<u>פתרון דינאמי</u>: נסתכל על האיבר האחרון a_n. או שהוא חלק מהפתרון או שלא. תהי $A^* = \left(a_{i_1}, a_{i_2}, \dots, a_{i_{n-1}}\right)$ תת-סדרה ב"ת גדולה ביותר.

יש שתי אפשרויות: $i_k = n$ או $i < n$.

אם $i_k < n$ אז a_n לא חלק מופיע ב-A^*, ובמקרה זה A^* היא הפתרון האופטימלי עבור $(a_1, \dots, a_{n-1})$.

אם $i_k = n$, אז $i_{k-1} \leq n - 2$ כי A^* היא ב"ת. לכן $\left(a_{i_1}, a_{i_2}, \dots, a_{i_{n-1}}\right)$ היא תת-סדרה ב"ת גדולה ביותר עבור $(a_1, \dots, a_{n-2})$.

נעיר ש- $\left(a_{i_1}, a_{i_2}, \dots, a_{i_{n-1}}\right)$ חייב להיות פתרון אופטימלי ל- $n-2$ האיברים הראשונים של A, כי אם לא כך, והיה פתרון אחר, נקרא לו B, היינו מצרפים את a_n ל-B ובכך מקבלים פתרון B^* שהוא יותר טוב מ- A^*, בסתירה לאופטימליות של A^*.

<u>פתרון רקורסיבי</u>: נסמן ב-$f(1)$ את הערך הגדול ביותר של תת-סדרה ב"ת מתוך $(a_1, \dots, a_i)$. המטרה שלנו היא לחשב את $f(n)$.

$$f(i) = \begin{cases} 0 & i = 0 \\ \max\{a_i, 0\} & i = 1 \\ \max\{f(i-1), f(i-2) + a_i\} & i \geq 2 \end{cases}$$

<u>הוכחת נכונות</u>:

נראה שהערך המוחזר - $f(n)$ מחזיק את הערך הגדול ביותר של כל תתי הסדרות הב"ת.

<u>למה</u>: לכל מספר שלם $0 \leq k \leq n$, הפונקציה $f(k)$ מחזירה ערך גדול ביותר של תת סדרה $(a_1, \dots, a_k)$.

<u>הוכחה</u>: באינדוקציה על k.

ישנם שני מקרי בסיס: $k = 0$ - כאן תת הסדרה היא ריקה וערכה 0.
$k = 1$ - כאן יש ערך בודד אם $a_1 > 0$, או שלא לוקחים איברים כלליים.

<u>צעד אינדוקציה</u>: נניח נכונות עבור $k < i$, ונוכיח עבור $k = i$.

נב"ש שיש $f(i)$ שאינה מחזירה את הערך המקסימלי. נסמן את הערך האופטימלי האמיתי ב-f^*. ישנם שני מקרים:

1. a_i מופיע בפתרון האופטימלי. במקרה הזה:

$$* \; f^* > f(i) = \max\{f(i-1), a_i + f(i-2)\} \geq a_1 + f(i-2)$$

כעת $f^* - a_i$ מייצג תת-סדרה מתוך $a_1, a_2, \dots, a_{i-2}$.

מהנחת האינדוקציה על i, אנו יודעים ש $f(i-2)$ הוא הפתרון האופטימלי עבור $a_1, a_2, \dots, a_{i-2}$.

לכן

$$** \; f^* - a_i \leq f(i-2)$$

אבל לפי *, מתקיים $f^* - a_i > f(i-2)$, וזו סתירה ל-**.

2. a_i לא מופיע בפתרון האופטימלי. במקרה זה:

$$* \; f^* > f(1) = \max\{f(i-1), a_i + f(i-2)\} \geq f(i-1)$$

כאן f^* הוא ערך של תת-סדרה ב"ת מתוך $a_1, a_2, \dots$. מהנחת האינדוקציה נובע ש-$f(i-1)$ הוא הפתרון האופטימלי עבור $a_1, a_2, \dots, a_{i-1}$.

לכן:

$$***\ f^* < -f(i-1)$$

וזו סתירה ל-*.

ניתן לחשב את תת הסדרה האופטימלית.

נקצה מערך חדש $D[0,1,\dots,n]$.

$D[1]$ מכילה שתי אפשרויות – האם a_i נבחר או לא נבחר לפתרון האופטימלי.

$F = F[0,1,\dots,n]$ מערך בעל $n+1$ כניסות.

<u>מטרה</u>: לחשב את $f(i)$ ולאחסנו ב-$F(i)$.

```
F[0]=0
F[1] = max{0,a1}
For i<2 to n:
    F[i]=max{F[i-1], ai+F[i-2]}
    if F[i]=F[i-1]:
        D[i] = 1
    else:
        D[i]=0
Return F[n], D[0,…,n]
```

הרצאה 5

<u>עצים פורשים מינימום</u>

יהי $G = (V, E)$ מולטי גרף קשיר, ממושקל עם פונקציות $w: E \to \mathbb{R}$. יהי $T = (V, E_T)$ עץ פורש של G. אז נאמר שהמשקל של T הוא:

$$w(T) = \sum_{e \in E_T} w(e)$$

עץ פורש מינימום של G הוא עץ פורש שמשקלו קטן ביותר.

<u>הנחות</u>: אין לולאות עצמיות (קשת מקודקוד לעצמו), אין קשתות מקבילות (כמה קשתות בין שני קודקודים).
יכולים להיות משקלים שליליים, אך נתייחס אליהם בהמשך.

<u>הגדרה</u>: תהי $\emptyset \neq S \subset V$. החלוקה $(S, V \backslash S)$, נקראת חתך של G.
הקשתות $e \in \{(u,v) | \; u \in S, v \in V \backslash S\}$ נקראות קשתות שחוצות את החתך.

<u>בחירה חמדנית</u>

$G = (V, E)$ גרף לא מכוון עם פונקציית משקל $w: E \to \mathbb{R}$. יהי $(S, V \backslash S)$ חתך של G. קשת $(u, v) \in E$ שחוצה את החתך נקראת קשת קלה ביותר, אם היא מינימלית במשקל מבין הקשתות החוצות את החתך.

<u>למת הבחירה החמדנית</u>: יהי G גרף קשיר וממושקל, עם פונקציית משקל $w: E \to \mathbb{R}$.
לכל תת קבוצה לא ריקה $S \subset V$, ולכל קשת e שהיא הקלה ביותר בחתך $(S, V \backslash S)$ קיים עפ״מ (עץ פורש מינימום), שמכיל את e.

<u>הוכחה</u>: יהי $(S, V \backslash S)$ חתך ב-G. ותהי $e = (u, v)$ קשת קלה ביותר בחתך $(S, V \backslash S)$, כאשר $u \in S, v \in V \backslash S$.

יהי $T = (V, E_T)$ עפ״מ עבור G. אם $e \in T$ סיימנו. אחרת, נבנה עפ״מ שמכיל אותו:

מכיוון ש-T עץ פורש, חייב להיות בו מסלול פשוט p שמחבר את u, v. מההנחה, $e \notin p$.

מכיוון ש-p מתחיל מצד אחד S ומסיים בצד שני $V \backslash S$, חייבת להיות לפחות קשת אחת שחוצה את החתך. תהי $e' = (u', v')$ הקשת הראשונה על p שחוצה את החתך, כאשר הולכים מ-u ל-v. נשים לב כי $u' \in S, v' \in V \backslash S$.

ניתן לכתוב את p כך: $p = p_1 \to e' \to p_2$. נבנה גוף חדש $T' = (V, E_{T'})$, כאשר $E_{T'} = (E_T \cup \{e\}) \backslash \{e'\}$.

כלומר מסירים את e' ומוסיפים את e. כעת נראה ש-T' הוא עפ״מ.

לשם כך נראה:

1) T' עץ פורש – קשיר וחסר מעגלים.
2) המשקל של T' חייב להיות קטן/שווה למשקל של T.

T' עץ פורש:

$$|E_{T'}| = |E_T| + 1 - 1 = |E_T| = |V| - 1$$

הוא גם קשיר. לכל שני קודקודים x, y, נסתכל על המסלול בינהם ב-T. נסמנו p'. אם $e' \notin p'$, אז המסלול נמצא גם ב-T'. אחרת, נסתכל על המסלולים בין x ל-u, ובין y ל-v. נסמנם p_1, p_2.
כעת נוכל להשתמש בצלע שהוספנו $e = (u, v)$. נקבל מסלול $p_1 \to e \to p_2$. כנדרש.

<u>המשקל של T':</u>

נראה ש-$w(T')$ הוא קטן ביותר.

מכיוון שהנחנו ש-e קלה ביותר, $w(e) \le w(e')$. לכן:

$$w(T') = w(T) - w(e') + w(e) \le w(T)$$

מצד שני, כיוון ש T הוא עפ"מ, אז $w(T) \le w(T')$. לכן $w(T) = w(T')$. וכמובן ש-$e \in T'$.

<u>כיווץ קשתות</u>

בהינתן מולטי-גרף $G = (V,E)$ חסר לולאות, <u>מיזוג</u> קודקודים u,v לקודקוד אחד uv מוביל לכיווץ הקשת (u,v).

כאן נקבל מולטי-גרף חדש, $G_{\setminus e} = \left(V_{\setminus e}, E_{\setminus e}\right)$, כאשר:

$$V_{\setminus e} = V - \{u,v\} \cup \{uv\}$$

הגדרה של פונקציית הכיווץ:

$$f: V \to V_{\setminus e}$$

$$f(x) = \begin{cases} x & x \neq u,v \\ uv & x = u \ or \ x = v \end{cases}$$

$$E_{\setminus e} = \left\{\left(f(x), f(y)\right) \mid (x,y) \in E\right\}$$

אם $e' = (u,w) \in E$, אז הקשת משתנה ל- (uv,w) ב-$E_{\setminus e}$.

מהו $\left(f(u), f(v)\right)$? זה (uv,uv) – זוהי לולאה עצמית.

בתהליך הזה נסיר את הלולאה (uv,uv) מ-$G_{\setminus e}$. בהינתן מושג הכיווץ ובעזרת למת הבחירה החמדנית, נוכל לייצר אלגוריתם גנרי שבונה עפ"מ.

$E_T = \emptyset$

While $|V| > 1$:

  Let e be the min weight edge in some cut $(S, V \setminus S)$ of G.

  Add e to E_T.

  Contract e.

Return E_T.

<u>למה 2 – תכונת תת המבנה האופטימלי</u>

יהי $G = (V,E)$ מולטי-גרף קשיר חסר לולאות, עם פוני' משקל $w: E \to \mathbb{R}$. יהי $(S, V \setminus S)$ חתך ב-G.
תהי $e = (u,v)$ קשת קלה ביותר בחתך $(S, V \setminus S)$. יהי $T' = \left(V_{\setminus e}, E_{T'}\right)$ עפ"מ עבור $G_{\setminus e}$.

יהי $E_T = E_{T'} \cup \{e\}$. אזי $T = (V, E_T)$ הוא עפ"מ עבור G.

<u>הוכחה</u>: נראה:

1) T עץ פורש.
2) משקלו של T מינימלי.

<u>T עץ פורש</u>: יש להראות ש-T קשיר וגם $|E_T| = |V| - 1$.

היות ש- $\left|V_{\setminus e}\right| = |V| - 1$ וגם $|E_{T'}| = \left|V_{\setminus e}\right| - 1$, נקבל:

$$|E_T| = |E_{T'}| + 1 = \left|V_{\setminus e}\right| = |V| - 1$$

כעת עבור קשירות: עבור זוג קודקודים $x,y \in V$, נסתכל על המסלול המחבר בין $f(x), f(y)$ ב-T'. אם המסלול לא מכיל את הקודקוד uv, אז אותו מסלול נמצא גם ב-T.

אחרת, אם uv מופיע על המסלול אז נחליף אותו בקשת $e = (u, v)$. אכן נקבל מסלול חוקי ב-T.

לכן T חייב להיות קשיר, ומהעובדה ש- $|E_T| = |V| - 1$, נובע שהוא עץ פורש.

<u>משקל מינימלי</u>: נניח שמשקל T אינו מינימלי. יהי $\hat{T} = (\hat{V}, \hat{E})$ עפ״מ ב-G.

מלמה 1, e חייב להיות ב- $\hat{T}$.

יהי $\hat{T}' = (V_{\setminus e}, E_{\hat{T}'})$, כאשר $E_{\hat{T}'} = E_{\hat{T}} \setminus \{e\}$.

נשים לב ש $\hat{T}'$ הוא עפ״מ ב-$G_{\setminus e}$, כי:

$$|E_{\hat{T}'}| = |E_{\hat{T}}| - 1 = |V| - 1 - 1 = |V_{\setminus e}| - 1$$

ולכל זוג קודקודים $x, y \in V_{\setminus e}$, אם המסלול בינהם p הנ״ל הכיל את e, אז $p_{\setminus e}$ הוא מסלול מחבר בינהם ב-$\hat{T}'$. אם $e \in p$, אז אותו מסלול בין x ל-y משתמר ב-$\hat{T}'$.

כמו כן, $E_{\hat{T}'} \subseteq E$ ולכן:

$$w(\hat{T}') = w(\hat{T}) - w(e) < w(T) - w(e) = w(T') + w(e) - w(e) = w(T')$$

לכן $w(\hat{T}') < w(T')$, בסתירה למינימליות של T'.

הרצאה 6

-- חלק ראשון אלגוריתם למציאת עפ"מ חסר --

<u>מסלולים קצרים ביותר</u>

$G=(V,E)$ גרף (יכול להיות מכוון או לא מכוון). $w: E\to\mathbb{R}$ פונקציית משקל.

המטרה שלנו תהיה למצוא את המסלול הקצר ביותר בין שני קודקודים. מה הכוונה "מסלול קצר ביותר"?

1. אם הגרף לא ממושקל, אז הכוונה היא למספר הקשתות במסלול.
2. אם הגרף ממושקל, הכוונה היא לסכום המשקלים של הקשתות במסלול.

<u>סימון</u>: נסמן את העלות של המסלול הקצר ביותר בין זוג קודקודים u ל-v בתור: $\delta(u,v)$.
אם לא קיים מסלול בינהם $\Leftarrow \delta(u,v)=\infty$.

בקורס נציג 3 גרסאות עיקריות:

1. Single pair – ננסה למצוא מסלול קצר ביותר בין זוג קודקודים נתון $v,u\in V$.
2. Single source – ננסה למצוא מסלולים קצרים ביותר מקודקוד s ליתר הקודקודים $v\in V$.
3. All pairs – מציאת מסלולים קצרים ביותר בין כל זוגות הקודקודים $u,v\in V$.

אורך מקסימלי של מסלול הוא $|V|-1$.

בבעית ה-single source יש $|V|-1$ מסלולים ולכן ייצוג נאיבי עלול לעלות Type equation here.

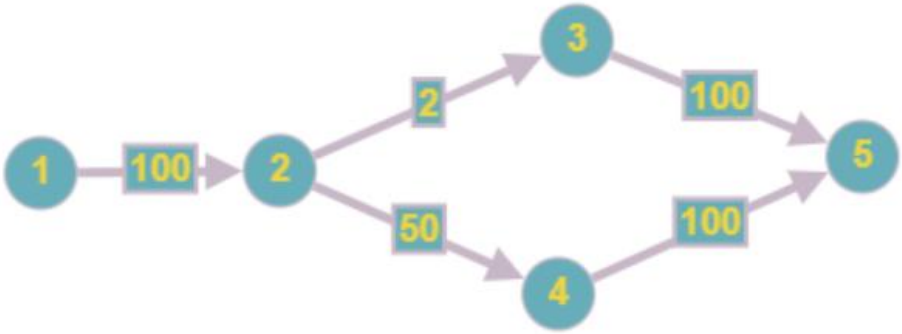
הרצאה 11 – רשתות זרימה

בהרצאה הזו אנחנו נתמקד ברשתות זרימה, שמיוצגות על ידי גרפים.

בפרט, נגדיר רשת זרימה בתור גרף מכוון G עם פונקציית משקל על הקשתות $w: E \to \mathbb{R}$. המשקל מייצג את הקיבולת של הקשת בפרק זמן מסויים.

בדרך כלל נסמן את קודקוד המקור בתור s ואת קודקוד הניקוז בתור t.

דוגמא: הקיבולת של הגרף הבא היא 52.



פונקציית הקיבולת

נוכל להגדיר פונקציית קיבולת $c: E \to \mathbb{R}^+$ שמסמלת את הקיבולת של כל קודקוד.

אם קשת בין שני קודקודים לא קיימת, אז הקיבולת בקשת בינהם מוגדרת להיות 0.

ערך זרימה

ערך הזרימה של גרף G מסומן לרוב ב-$|f|$. הוא מסמל את כמות המידע שעוברת מקודקוד המקור לקודקוד הניקוז.

$$|f| = \sum_{v \in V} f(s, v)$$

ההגדרה הזאת באה עם כמה תנאים:

1. הזרימה אינה גדולה מהקיבולת: $\forall u, v \in V: 0 \leq f(u, v) \leq c(u, v)$.
2. הזרימה נשמרת: $\sum_{v \in V} f(u, v) = \sum_{w \in V} f(v, w)$ לכל $v \in V$. כלומר כל קודקוד מוציא את אותה כמות הזרימה כפי שהוא מכניס.

הגדרה אלטרנטיבית

ניתן להגדיר גם את ערך הזרימה בתור:

$$f: V \times V \to \mathbb{R}$$

כאשר:

1. סימטריה: $f(u, v) = -f(v, u)$.
2. זרימה קטנה שווה לקיבולת: $f(u, v) \leq c(u, v)$.
3. שימור זרימה: לכל $v \in V \setminus \{s, t\}$, מתקיים: $\sum_{u \in V} f(u, v) = 0$.

תכונות בסיסיות של זרימה

יהיו $X, Y \subseteq V$ תתי קבוצות של קודקודים. נגדיר:

$$f(X, Y) = \sum_{x \in X} \sum_{y \in Y} f(x, y)$$

טענה 4

תהי f זרימה בגרף $G=(V,E)$. אזי:

1. $\forall X\subseteq V: f(X,X)=0$.
2. $\forall X,Y\subseteq V: f(X,Y)=-f(Y,X)$.
3. לכל $X,Y,Z\subseteq V$ כך ש $X\cap Y=\emptyset$, אזי: $f(X\cup Y,Z)=f(X,Z)+f(Y,Z)$.
   בנוסף: $f(Z,X\cup Y)=f(Z,X)+f(Z,Y)$.

למה 5

תהי f זרימה ברשת זרימה $G=(V,E)$. מתקיים:

$$|f|=\sum_{v\in V}f(v,t)=f(V,t)$$

הוכחה: מתכונת הסימטריה, נובע ש:

$$f(V,V\backslash\{s,t\})=-f(V\backslash\{s,t\},V)=-\sum_{x\in V\backslash\{s,t\}}\sum_{y\in V}f(x,y)=-\sum_{x\in V\backslash\{s,t\}}0=0$$

$$\Rightarrow f(V,V\backslash\{s,t\})=0$$

ניתן להביע $V=\{s\}\cup\{V\backslash\{s\}\}$. נשים לב כי $\{s\},\{V\backslash\{s\}\}$ קבוצות זרות. לכן מטענה 4:

$$f(V,V)=f(s,V)+f(V\backslash\{s\},V)$$

ומכיוון ש: $V\backslash\{s\}=\{t\}\cup\{V\backslash\{s,t\}\}$ נקבל:

$$f(V,V\backslash\{s\})=f(V,V\backslash\{s,t\})+f(V,t)$$

$$|f|=f(s,V)=\underbrace{f(V,V)}_{0}-f(V\backslash\{s\},V)=f(V,V\backslash\{s\})$$

$$f(V,V\backslash\{s\})=\underbrace{f(V,V\backslash\{s,t\})}_{0}+f(V,t)=f(V,t)=|f|$$

חתך (s,t)

חתך (s,t) הוא חתך $(S,V\backslash S)$. נקרא לו גם (S,T) לשם פשטות, כאשר $T=V\backslash S$, ומתקיים $s\in S$, $t\in T$ (קודקוד המקור וקודקוד הניקוד נמצאים בחלקים שונים של החתך).

<u>למה:</u> יהי $G=(V,E)$ רשת זרימה עם פונקציית קיבולת c, ותהי f זרימה ב-G. יהי $(S',V\backslash S')$ חתך (s,t) של G. אזי: $|f|=f(S,T)$.

<u>הוכחה:</u> נשים לב ש: $S'\cup T=V, T\cap S'=\emptyset$.

$$f(S\backslash\{s\},V)=\sum_{x\in S\backslash\{s\}}\sum_{y\in V}f(x,y)=*\sum_{x\in S\backslash\{s\}}0=0$$

* $S\backslash\{s\}$ לא מכילה את t ולכן הסכום מתאפס.

$$f(S,T)=f(S,V)-\underbrace{f(S,S)}_{0}=f(S,V)=f(S\backslash\{s\},V)+f(s,V)=f(s,V)=|f|$$

בעיית זרימת מקסימום

בהינתן רשת זרימה $G = (V,E)$ עם פונקציית קיבולת c, מצא זרימה ב-G בעלת ערך גדול ביותר מבין כל הזרימות האפשריות.

<u>גישה חמדנית</u>: כל עוד קיים מסלול מ-s ל-t שקשתותיו לא רוויות, נזרים עליו מה שאפשר.

רשת שיורית

הגדרה: הקיבולת השיורית היא:

$$c_f(u,v) = c(u,v) - f(u,v)$$

כלומר היא הקיבולת שנותרה לאחר שהזרמנו כבר משהו.

הרשת השיורית של G תחת f היא רשת זרימה $G_f = \left(V, E_f\right)$, שפונקציית הקיבולת שלה היא c_f, כאשר

$$E_f = \left\{(u,v) \middle| \ c_f(u,v) > 0\right\}$$

כלומר אנחנו מסתכלים על כל הקשתות שהקיבולת השיורית שלהן גדולה מ-0.

<u>Ford-Fulkerson algorithm</u>