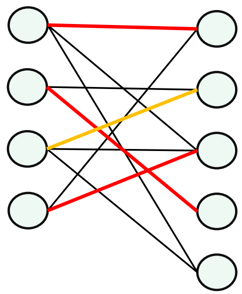
מבני נתונים ואלגירותמים 2
תירגול: שידוך מקסימום
וכיסוי בצמתים ממושקל בגרף דו"צ

1

שידוך מקסימום בגרף דו"צ

- גרף דו-צדדי $G = (L \cup R, E)$:
  - קבוצת הצמתים של מחולקת לשני תתי קבוצות או צדדים L ו- R.
  - הקשתות שלו מחברות בין צמתים השייכים לצדדים שונים, כלומר $E \subseteq L \times R$.
- שידוך:
  - תת קבוצה של קשתות M שזרות בצמתים
  - כל צמת נוגע בלא יותר מקשת אחת מ- M.
- שידוך מקסימום:
  - שידוך שגודלו מירבי.
- שידוך מושלם:
  - שידוך M שמקיים $|M| = |L| = |R|$.



2

2

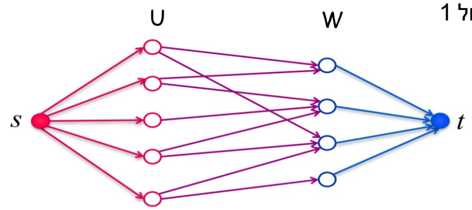
דוגמא

- לצער בעלי חיים הגיעו 100 משפחות במטרה לאמץ כלב. בכלבייה יש 100 כלבים לאימוץ ולכל משפחה רשימה של כלבים אותם היא מוכנה לאמץ.
- מצאו אלגוריתם שימצא את מספר האימוצים המקסימלי האפשרי.

3

דוגמא

- נייצג את המשפחות והכלבים בגרף דו צדדי $G = (U, W, E)$:
  - המשפחות יהיו בצד השמאלי של הגרף
  - הכלבים בצד הימני של הגרף
  - הקשתות ייצגו את רשימת הכלבים אותה כל משפחה מוכנה לאמץ ויהיו מכוונות משמאל לימין.
- בהינתן הגרף G נבנה רשת זרימה H באופן הבא:
  - נוסיף קודקוד S וניצור קשתות ממנו לכל קודקודי U
  - נוסיף קודקוד t וניצור קשתות אליו מכל קודקודי W
  - ניתן לכל הקשתות קיבול 1



4

דוגמא

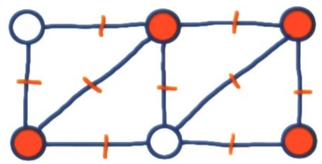
- קיבלנו רשת זרימה, נריץ עליה אלג' למציאת זרימה מקסימאלית מ-S ל-t
- כפי שהוכח בהרצאה מתקיים שאם f היא זרימה שלמה ב-H, אז קיים שידוך ב- G כך ש- $|M| = |f|$.

5

5

בעיית הכיסוי בצמתים

- יהי G=(V,E), גרף לא מכוון. $U \subseteq V$ נקרא כיסוי בצמתים (Vertex Cover) של הגרף אם לכל קשת $(u,v) \in E$ מתקיים $u \in U$ או $v \in U$.
- במילים אחרות, סילוק U מהגרף מותיר אותו חסר קשתות.



6

6

הגדרת הבעיה

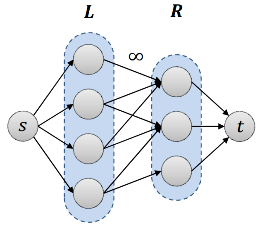
- הבעיה: בהינתן גרף עם משקלים על הצמתים מצא VC במשקל מינימלי.
- הערות:
  1. ניתן להניח שהמשקלים אי-שליליים
  2. בגרף כללי הבעיה היא NP-קשה
- נפתור את בעיית הכיסוי בצמתים בגרף דו צדדי.
- נשתמש בזרימה על מנת לפתור את הבעיה.

7

7

שלב 1- בניית רשת N

- נגדיר את הגרף המכוון $G' = (V', E')$ באופן הבא:
- $V' = V \cup \{s,t\}$
- E' כולל את הקשתות:
- לכל $u \in L$ נוסיף קשת מ-s ל-u כך ש-$c(s,u)=w(u)$
- לכל $v \in R$ נוסיף קשת מ-v ל-t כך ש-$c(v,t)=w(v)$
- נכוון כל קשת $(u,v) \in E$ מ-u ל-v ונגדיר ש- $c(u,v)=\infty$
- $E' = \{su \mid u \in L\} \cup \{vt \mid v \in R\} \cup \{uv \mid uv \in E, u \in L, v \in R\}$



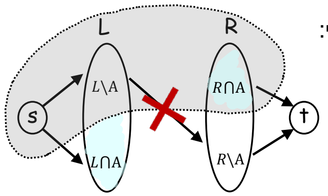
8

8

טענה 1

- כיסוי בצמתים A בגרף G מגדיר חתך $(S, \bar{S})$ שקיבולו w(A) .

הוכחה:



- בהינתן כיסוי בצמתים A נגדיר חתך (S, S) ע"י:
  - $S = (L \setminus A) \cup (R \cap A) \cup \{s\}$
- נתבונן בחתך (S, S), קשתות החתך כוללות:
  1. קשתות מ- s ל-$A \cap L$
  2. קשתות מ-$A \cap R$ ל-t
- לא תתכננה קשתות מ-L\A ל-R\A כי אחרת A אינו כיסוי.
- לכן

$$C(S, S) = \sum_{u \in A \cap L} c(s, u) + \sum_{v \in A \cap R} c(v, t) =$$
$$= \sum_{u \in A \cap L} w(u) + \sum_{v \in A \cap R} w(v) = w(A)$$

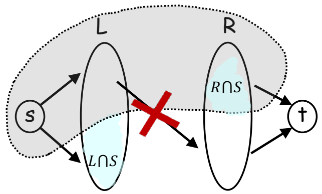
9

9

טענה 2

- כל חתך (S, S) בעל קיבול סופי מגדיר כיסוי בצמתים A במשקל $C(S, S)$.

הוכחה:



- בהינתן חתך (S, S) נגדיר כיסוי A ע"י:
  - $A = (S \cap L) \cup (S \cap R)$
- כיוון שהחתך בעל קיבול סופי הוא אינו מכיל קשתות מ-L ל-R לכן אין קשת מ-$S \cap L$ ל-$S \cap R$
- לכן A מהווה כיסוי בצמתים.
- כמו כן

$$w(A) = w(A \cap L) + w(A \cap R) = \sum_{u \in A \cap L} w(u) +$$
$$+ \sum_{v \in A \cap R} w(v) = \sum_{u \in A \cap L} c(s, u) + \sum_{v \in A \cap R} c(v, t) = C(S, S)$$

10

10

האלגוריתם למציאת VC מינימום

- קלט: גרף דו צדדי $G=(L \cup R, E)$.
- פלט: כיסוי בצמתים A של הגרף G.

1. בנה את רשת הזרימה N כפי שהוגדר בשקפים הקודמים.
2. הרץ את האלגוריתם Ford-Fulkerson לחישוב זרימת מקסימום ב-N .
3. מצא חתך מינימום (לפי משפט ה- Min-Cut Max-Flow)
4. החזר את הקבוצה: $A = (\bar{S} \cap L) \cup (S \cap R)$

11

11

מסקנה

- <u>משקל VC מינימום = קיבול חתך מינימום</u>
- לכן כדי לפתור את הבעיה נמצא זרימה מקסימלית ברשת ומתוכה נמצא חתך מינימום, וממנו נחלץ את ה-VC המתאים.
- <u>הערות:</u>
- במקום להשתמש ב-∞ לקיבול קשת (u,v) מ-L ל-R נוכל להשתמש בערך $\min\{w(u),w(v)\}+1$.
- קיבול זה מבטיח שהקשת לא תהיה רוויה לעולם, ולכן קבוצת הזרימות החוקיות לא תשתנה.
- בבעיה הלא ממושקלת הרשת זהה לרשת שבנינו עבור בעיית השידוך מקסימום ולכן נקבל זמן ריצה של $O(\sqrt{n} \cdot m)$

12

12

מבני נתונים ואלגוריתמים 2
תירגול: Fast matrix multiplication

Based on: Introduction to Algorithms
by Cormen, Leiserson, Rivest, and Stein

13

13

הכפלת מטריצות בסיסית

- נרצה להכפיל שני מטריצות מגודל 2×2 לדוגמה
: $C = A \times B$

$$\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

$$c_{11} = a_{11}b_{11} + a_{12}b_{21} \qquad c_{21} = a_{21}b_{11} + a_{22}b_{21}$$

$$c_{12} = a_{11}b_{12} + a_{12}b_{22} \qquad c_{22} = a_{21}b_{12} + a_{22}b_{22}$$

- ניתן לבצע הכפלת מטריצות מגודל 2x2 עם 8 פעולות כפל.

14

14

הכפלת מטריצות בסיסית

Algorithm

```
void matrix_mult (){
  for (i = 1; i <= N; i++) {
       for (j = 1; j <= N; j++) {
             compute Ci,j;
       }
}}
```

Time analysis

$$C_{i,j} = \sum_{k=1}^{N} a_{i,k} b_{k,j}$$

$$\text{Thus } T(N) = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} c = cN^3 = O(N^3)$$

15

15

אלגוריתם הפרד-ומשול פשוט להכפלת מטריצות

- נשתמש באלגוריתם הפרד-ומשול לחשב את תוצאת המטריצה $C = A \cdot B$
- יהיו A ו B שני מטריצות מגודל $n \times n$, כאשר n חזקה של 2
- נחלק את A,B ו C לארבע מטריצות מגודל $n/2 \times n/2$

$$\left(\begin{array}{c|c} c_{11} & c_{12} \\ \hline c_{21} & c_{22} \end{array}\right) = \left(\begin{array}{c|c} a_{11} & a_{12} \\ \hline a_{21} & a_{22} \end{array}\right) \left(\begin{array}{c|c} b_{11} & b_{12} \\ \hline b_{21} & b_{22} \end{array}\right)$$

- בכל אחד מארבעת המכפלות הנ"ל מופיעים שני מכפלות של מטריצות מגודל $n/2 \times n/2$ וחיבור של התוצאות מגודל $n/2 \times n/2$

$$c_{11} = a_{11}b_{11} + a_{12}b_{21} \qquad c_{21} = a_{21}b_{11} + a_{22}b_{21}$$

$$c_{12} = a_{11}b_{12} + a_{12}b_{22} \qquad c_{22} = a_{21}b_{12} + a_{22}b_{22}$$

16

16

אלגוריתם הפרד-ומשול פשוט להכפלת מטריצות

```
SQUARE-MATRIX-MULTIPLY-RECURSIVE(A, B)
n = A.rows
let C be a new n × n matrix
if n == 1
    c11 = a11 · b11
else partition A, B, and C as in equations (4.9)
    C11 = SQUARE-MATRIX-MULTIPLY-RECURSIVE(A11, B11)
          + SQUARE-MATRIX-MULTIPLY-RECURSIVE(A12, B21)
    C12 = SQUARE-MATRIX-MULTIPLY-RECURSIVE(A11, B12)
          + SQUARE-MATRIX-MULTIPLY-RECURSIVE(A12, B22)
    C21 = SQUARE-MATRIX-MULTIPLY-RECURSIVE(A21, B11)
          + SQUARE-MATRIX-MULTIPLY-RECURSIVE(A22, B21)
    C22 = SQUARE-MATRIX-MULTIPLY-RECURSIVE(A21, B12)
          + SQUARE-MATRIX-MULTIPLY-RECURSIVE(A22, B22)
return C
```

17

17

שיטת האב (המאסטר) - תזכורת

Let $a \geq 1$ and $b > 1$ be constants, let $f(n)$ be a function, and let $T(n)$ be defined on the nonnegative integers by the recurrence

$$T(n) = aT(n/b) + f(n) ,$$

where we interpret n/b to mean either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then $T(n)$ has the following asymptotic bounds:

1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$. ■

18

18

ניתוח סיבוכיות

- נכתוב את משוואת הרקורסיה להכפלת מטריצות:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1\ , \\ 8T(n/2) + \Theta(n^2) & \text{if } n > 1\ . \end{cases}$$

- לפי שיטת המאסטר :
  - $a=8,\ b=2,\ f(n) \in O(n^2)$
  - $\log_2 8 = 3 \rightarrow f(n) \in O(n^3) \rightarrow T(n) \in O(n^3)$

Strassens's Matrix Multiplication

- בשנת 1969 סטרסן הראה כי ניתן לבצע כפל מטריצות מגודל 2x2 באמצעות 7 פעולות כפל ו - 18 פעולות חיבור או חיסור.

$$\begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} \begin{pmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{pmatrix} = \begin{pmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{pmatrix} = \begin{pmatrix} m_1+m_4-m_5+m_7 & m_3+m_5 \\ m_2+m_4 & m_1+m_3-m_2+m_6 \end{pmatrix}$$

$m_1 = (a_{00}+a_{11})*(b_{00}+b_{11})$

$m_2 = (a_{10}+a_{11})*b_{00}$

$m_3 = a_{00}*(b_{01}-b_{11})$

$m_4 = a_{11}*(b_{10}-b_{00})$

$m_5 = (a_{00}+a_{01})*b_{11}$

$m_6 = (a_{10}-a_{00})*(b_{00}+b_{01})$

$m_7 = (a_{01}-a_{11})*(b_{10}+b_{11})$

Strassens's algorithm

1. מטריצות הקלט A ו B וכן מטריצת התוצאה C יחולקו לתתי מטריצות מגודל $n/2 \times n/2$

- סיבוכיות זמן: $O(1)$ לחישובי אינדקסים

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

Each block is (n/2)×(n/2)

21

21

Strassens's algorithm

2. צרו 10 מטריצות $S1, S2, \ldots, S10$ מגודל $n/2 \times n/2$, כאשר כל אחת מהן היא סכום או חיסור של שניים מהמטריצות שחושבו בשלב 1.

- סיבוכיות זמן: $O(n^2)$ לחישוב כל 10 המטריצות

$$\begin{aligned}
S_1 &= B_{12} - B_{22}, \\
S_2 &= A_{11} + A_{12}, \\
S_3 &= A_{21} + A_{22}, \\
S_4 &= B_{21} - B_{11}, \\
S_5 &= A_{11} + A_{22}, \\
S_6 &= B_{11} + B_{22}, \\
S_7 &= A_{12} - A_{22}, \\
S_8 &= B_{21} + B_{22}, \\
S_9 &= A_{11} - A_{21}, \\
S_{10} &= B_{11} + B_{12}.
\end{aligned}$$

22

22

Strassens's algorithm

3. ע"י שימוש בתתי המטריצות משלב 1, ו 10 המטריצות שחושבו בשלב 2, באופן רקורסיבי יחושבו 7 מכפלות של מטריצות לתוך P1,P2,..P7.

כל מטריצת תוצאה Pi היא מגודל n/2 x n/2

$$
\begin{aligned}
P_1 &= A_{11} \cdot S_1 &&= A_{11} \cdot B_{12} - A_{11} \cdot B_{22} ,\\
P_2 &= S_2 \cdot B_{22} &&= A_{11} \cdot B_{22} + A_{12} \cdot B_{22} ,\\
P_3 &= S_3 \cdot B_{11} &&= A_{21} \cdot B_{11} + A_{22} \cdot B_{11} ,\\
P_4 &= A_{22} \cdot S_4 &&= A_{22} \cdot B_{21} - A_{22} \cdot B_{11} ,\\
P_5 &= S_5 \cdot S_6 &&= A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22} ,\\
P_6 &= S_7 \cdot S_8 &&= A_{12} \cdot B_{21} + A_{12} \cdot B_{22} - A_{22} \cdot B_{21} - A_{22} \cdot B_{22} ,\\
P_7 &= S_9 \cdot S_{10} &&= A_{11} \cdot B_{11} + A_{11} \cdot B_{12} - A_{21} \cdot B_{11} - A_{21} \cdot B_{12} .
\end{aligned}
$$

23

23

Strassens's algorithm

4. תתי המטריצות C11,C12,C21,C22 של מטריצת התוצאה C יחושבו ע"י קומבינציות שונות של פעולות חיבור וחיסור של מטריצות Pi.

- סיבוכיות זמן: $O(n^2)$ לחישוב ארבעת תתי המטריצות

$$
\begin{aligned}
C_{11} &= P_5 + P_4 - P_2 + P_6 .\\
C_{12} &= P_1 + P_2 ,\\
C_{21} &= P_3 + P_4\\
C_{22} &= P_5 + P_1 - P_3 - P_7 ,
\end{aligned}
$$

24

24

Strassen Algorithm

Strassen(A, B)

if $A.length == 1$ **then**
  return $A[1] \cdot B[1]$
end if
Let C be a new n by n matrix
$A11 = A[1..n/2][1..n/2]$
$A12 = A[1..n/2][n/2+1..n]$
$A21 = A[n/2+1..n][1..n/2]$
$A22 = A[n/2+1..n][n/2+1..n]$
$B11 = B[1..n/2][1..n/2]$
$B12 = B[1..n/2][n/2+1..n]$
$B21 = B[n/2+1..n][1..n/2]$
$B22 = B[n/2+1..n][n/2+1..n]$
$S_1 = B12 - B22$
$S_2 = A11 + A12$
$S_3 = A21 + A22$
$S_4 = B21 - B11$
$S_5 = A11 + A22$

25

25

Strassen Algorithm

$S_6 = B11 + B22$
$S_7 = A12 - A22$
$S_8 = B21 + B22$
$S_9 = A11 - A21$
$S_{10} = B11 + B12$
$P_1 = Strassen(A11, S_1)$
$P_2 = Strassen(S_2, B22)$
$P_3 = Strassen(S_3, B11)$
$P_4 = Strassen(A22, S_4)$
$P_5 = Strassen(S_5, S_6)$
$P_6 = Strassen(S_7, S_8)$
$P_7 = Strassen(S_9, S_{10})$
$C[1..n/2][1..n/2] = P_5 + P_4 - P_2 + P_6$
$C[1..n/2][n/2+1..n] = P_1 + P_2$
$C[n/2+1..n][1..n/2] = P_3 + P_4$
$C[n/2+1..n][n/2+1..n] = P_5 + P_1 - P_3 - P_7$
return C

26

26

ניתוח סיבוכיות

- נכתוב את משוואת הרקורסיה:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1\ , \\ 7T(n/2) + \Theta(n^2) & \text{if } n > 1\ . \end{cases}$$

- לפי שיטת המאסטר:
- a=7, b=2, f(n)∈ $O(n^2)$
- $\log_2 7 = 2.71$ → f(n)∈ $O(n^{2.71})$ → T(n)∈$O(n^{2.71})$

27

27

דוגמא 1

$$\begin{pmatrix} 1 & 3 \\ 7 & 5 \end{pmatrix}\begin{pmatrix} 6 & 8 \\ 4 & 2 \end{pmatrix}$$

השתמשו באלגוריתם סטרסן להכפלת המטריצות

$S_1 = 8 - 2 = 6$
$S_2 = 1 + 3 = 4$
$S_3 = 7 + 5 = 12$
$S_4 = 4 - 6 = -2$
$S_5 = 1 + 5 = 6$

$S_6 = 6 + 2 = 8$
$S_7 = 3 - 5 = -2$
$S_8 = 4 + 2 = 6$
$S_9 = 1 - 7 = -6$
$S_{10} = 6 + 8 = 14$

$P_1 = 6$
$P_2 = 8$
$P_3 = 72$
$P_4 = -10$

$P_5 = 48$
$P_6 = -12$
$P_7 = -84$

נקבל את מטריצת התוצאה

$C_{11} = 48 - 10 - 8 - 12 = 18$
$C_{12} = 6 + 8 = 14$
$C_{21} = 72 - 10 = 62$
$C_{22} = 48 + 6 - 72 + 84 = 66$

$$\begin{pmatrix} 18 & 14 \\ 62 & 66 \end{pmatrix}$$

28

28

דוגמא 2

- כיצד הייתם משנים את אלגוריתם סטרסן להכפלת מטריצות מגודל n x n עבור n שאינה חזקה של 2?

- הראו שסיבוכיות האלגוריתם היא T(n)= Θ($n^{\log 7}$).

29

29

דוגמא 2

- נרפד באפסים את מטריצות הקלט למטריצה בגודל חזקה שלמה של 2.

- הריפוד למספר הבא שהוא חזקה שלמה של 2 (נקרא לגודל זה m) לכל היותר יכפיל את הערך n

- נקבל: $m^{\lg 7} \leq (2n)^{\lg 7} = 7n^{\lg 7} \in O(n^{\lg 7})$

- וגם: $m^{\lg 7} \geq n^{\lg 7} \in \Omega(n^{\lg 7})$

- משני אי-השיוויונים נקבל: T(n)= Θ($n^{\log 7}$)

30

30