אלגוריתמים

מרצה
פרופ' אלכס סמורודניצקי

מתרגל | גלעד שטרן

דניאל דייצ'ב   דויד קיסר שמידט

deychev.com

אלגוריתמים | 67504

מרצה - ד"ר אלכס סמורדינצקי

דניאל דייצ'ב   דויד קיסר שמידט

12 באוגוסט 2022

מצאתם שגיאה? ספרו לנו! שלחו לנו מייל לאחת מהכתובות שכאן:
daniel.deychev@mail.huji.ac.il
david.keisarschm@mail.huji.ac.il

תוכן העניינים

רשימת אלגוריתמים

חלק I

הקדמה

בקורס זה נסקור בעיות חשובות ובאמצעותן נלמד איך לפתור בעיות אלגוריתמיות.

- ראשית נזהה את המידע שיש לנו.
- נבצע רדוקציה לבעיה מוכרת, כלומר נהפוך את הבעיה לבעיה שאנו יודעים לפתור.

לאחר מכן נשתמש בשיטות חשיבה אלגוריתמיות:

- פתרון איטרטיבי - נוסיף בכל איטרציה מידע נוסף ונמשיך לרוץ עד שנסיים. בשיטה זו אנו מגדירים פונקציית רווח מקסימלית, כמו שהוספנו צלע באלגוריתם קורסקל - חמדניים
- הפרד ומשול - נחלק את הבעיה שלנו לבעיות קטנות יותר - דינאמים
- נשתמש באקראיות - למשל, הטלת מטבע. נראה שפעמים רבות אין פתרון יעיל ללא אקראיות - הסתברותיים
- קירובים - יתכן שלבעיות מסוימות אין פתרון אופטימלי, ולכן ננסה לתת אלגוריתם יעיל שיתן פתרון מקורב. נלמד על בעיות NP קשות.

בחלק הראשון של הקורס נלמד על שיטות החשיבה.

דוגמה. נתונים מספקים שלמים $a_1 \ldots, a_n$ האם יש שניים מתוכם שמסתכמים ל-2021? נביט בשלושת הפתרונות הבאים:

1. כדי לפתור בעיה זו נוכל לעבור על כל הזוגות a_i, a_j ולבדוק האם $a_i + a_j = 2021$ ב-$\Theta\left(n^2\right) = \Theta\left(\binom{n}{2}\right)$.
2. נוכל למיין את המספרים בסדר עולה, ולכל $1 \leq i \leq n$ נבדוק האם $2021 - a_i$ נמצא ברשימה באמצעות חיפוש בינארי, סך הכל ב-$\Theta\left(n \log n\right)$.
3. פתרון קביל אחר הוא שימוש בטבלת גיבוב - נכניס את המספרים לטבלא ונבדוק עבור כל מספר a_i האם $2021 - a_i$ נמצא בטבלא, וזה יהיה $\Theta\left(n\right)$.

חלק II

אלגוריתמים חמדניים

אלגוריתם חמדני בונה פתרון באופן איטרטיבי כאשר בכל שלב הא ממקסם פונקציית רווח מקומית. נבנה תאוריה כללית המאפשרת לזהות דוגמאות של בעיות אלגוריתמיות שניתן לפתור אותן באמצעות אלגוריתם חמדן. נצבור בטחון עם דוגמאות רבות.

1 בעיית התרמיל השברי (Fractional Knapsack)

בעיה. גנב נכנס לחנות בה יש מספר פריטים. לכל פריט ערך ומשקל משלו. לגנב יש תרמיל שמוגרל במשקל המירבי שהוא יכול לשאת. הגנב מעוניין להכניס לתרמיל פריטים עם ערך כולל מקסימלי. הפריטים ניתנים לחלוקה.

קלט n - מספר הפריטים בחנות. $W \geq 0$ המשקל המירבי של התרמיל ורשימה של n זוגות של מספרים אי שליליים $\left(v_1, w_1\right), \ldots, \left(v_n, w_n\right)$ כאשר v_i הוא הערך הכולל של הפריט ה-i ו-w_i הוא המשקל הכולל של הפריט ה-i.

פלט רשימה של n מספרים $0 \le x_1 \dots, x_n \le 1$ כאשר x_i הוא חלקיות הפריט ה-i שנכנסת לתרמיל, המקיימת את אילוץ המשקל $\sum_{i=1}^{n} x_i w_i \le W$ וכך ש-$\sum_{i=1}^{n} x_i v_i$ מקסימלי.

דוגמה. $W = 50, n = 3$, $(60,10), (100,20), (120,30)$. נביט בפתרונות הבאים:

1. $x_1 = x_2 = 1, x_3 = 0$. אילוץ המשקל: $1{\cdot}30{+}1{\cdot}20{+}0{\cdot}10 = 50$ אכן מתקיים. הערך הכולל הינו $1{\cdot}120{+}1{\cdot}100{+}0{\cdot}60 = 220$.
2. נחשב לכל פריט את הערך הסגולי שלו - הערך ליחידת משקל $r_1 = \frac{120}{30} = 4, r_2 = 5, r_3 = 6$. נמיין את הערכים ונכניס כמה שיותר מפריטים עם ערך סגולי גבוה יותר. במקרה הזה, $x_3 = 1, x_2 = 1$ ו-$x_1 = \frac{50-20-10}{30} = \frac{2}{3}$. נבדוק את אילוץ המשקל: $\frac{2}{3} \cdot 30 + 1 \cdot 20 + 1 \cdot 10 = 50$. הערך הכולל הינו $\frac{2}{3} \cdot 120 + 1 \cdot 100 + 1 \cdot 60 = 240$ שהוא גבוה יותר מהפתרון הקודם.

פתרון. (הבעיה הכללית) נניח כי $\sum_{i=1}^{n} w_i \le W$ במקרה כזה נגדיר $x_1 = \ldots = x_n = 1$ וזהו בבירור פתרון חוקי ואופטימלי במקרה הזה.

עתה, נניח כי $\sum_{i=1}^{n} w_i > W$. נחשב לכל $1 \le i \le n$ את הערך הסגולי של הפריט ה-i על פי הנוסחא $r_i = \frac{v_i}{w_i}$ ונמיין את הפריטים בסדר יורד על פי הערכים הסגוליים שלהם. כלומר, מעכשיו נניח כי $r_1 \ge r_2 \ge \ldots \ge r_n$.

במקרה כזה קיים אינדקס $0 \le t \le n-1$ כך ש-$\sum_{i=1}^{t+1} w_i > W$. נגדיר $x_1 = x_2 = \ldots = x_t = 1$ ונגדיר $x_{t+1} = \frac{W - \sum_{i=1}^{t} w_i}{w_{t+1}}$ וגם $x_{t+2} = \ldots = x_n = 0$. נחזיר אם כך את $(x_1, x_2, \ldots, x_n)$.

טענה. האלגוריתם שנתנו מחזיר פתרון חוקי ואופטימלי לבעיה.

הוכחה: נבדוק את חוקיות הפלט ולאחר מכן אופטמיליות.

<u>חוקיות:</u> נבדוק כי $0 \le x_1, \ldots, x_n \le 1$. אכן $x_1 = \ldots = x_t = 1$ וגם $x_{t+2} = \ldots = x_n = 0$. עתה נבחין כי

$$0 = \frac{\sum_{i=1}^{t} w_i - \sum_{i=1}^{t} w_i}{w_{t+1}} \le x_{t+1} = \frac{W - \sum_{i=1}^{t} w_i}{w_{t+1}} < \frac{\sum_{i=1}^{t+1} w_i - \sum_{i=1}^{t} w_i}{w_{t+1}} = \frac{w_{t+1}}{w_{t+1}} = 1$$

שכן $w_{t+1} > W - \sum_{i=1}^{t} w_i$.

<u>אילוץ המשקל:</u> נחשב

$$\sum_{i=1}^{n} x_i w_i = \sum_{i=1}^{t} 1 \cdot w_i + x_{t+1} \cdot w_{t+1} + \sum_{j=t+2}^{n} 0 \cdot w_j = \sum_{i=1}^{t} w_i + \frac{W - \sum_{i=1}^{t} w_i}{w_{t+1}} \cdot w_{t+1} + 0$$
$$= \sum_{i=1}^{t} w_i + W - \sum_{i=1}^{t} w_i = W$$

נשים לב כי $\sum_{i=1}^{n} x_i w_i = W$ ובזאת הוכחנו את חוקיות הפתרון.

<u>אופטימליות:</u> יהי $(y_1,\ldots,y_n)$ פתרון חוקי כלשהו לבעיה. נוכיח כי $\sum_{i=1}^{n} y_i v_i \le \sum_{i=1}^{n} x_i v_i$. נוודא כי $\sum_{i=1}^{n} x_i v_i - \sum_{i=1}^{n} y_i v_i \ge 0$. נרשום

$$\begin{aligned}
\sum_{i=1}^{n} x_i v_i - \sum_{i=1}^{n} y_i v_i = \sum_{i=1}^{n} (x_i - y_i) v_i = \sum_{i=1}^{n} (x_i - y_i) w_i r_i & \\
&= \sum_{i=1}^{t} (x_i - y_i) w_i r_i + (x_{t+1} - y_{t+1}) w_{t+1} r_{t+1} + \sum_{j=t+2}^{n} (x_j - y_j) w_j r_j \\
&\ge \sum_{i=1}^{t} \left(\underbrace{x_i}_{1} - \underbrace{y_i}_{\le 1} \right) w_i r_{t+1} + (x_{t+1} - y_{t+1}) w_{t+1} r_{t+1} + \sum_{j=t+2}^{n} \left(\underbrace{x_j}_{0} - \underbrace{y_j}_{\ge 0} \right) w_j r_{t+1} \\
&= r_{t+1} \left(\sum_{i=1}^{t} (x_i - y_i) w_i + (x_{t+1} - y_{t+1}) w_{t+1} + \sum_{j=t+2}^{n} (x_j - y_j) w_j \right) \\
&= r_{t+1} \left(\sum_{i=1}^{n} (x_i - y_i) w_i \right) = r_{t+1} \left(\sum_{i=1}^{n} x_i w_i - \sum_{i=1}^{n} y_i w_i \right) \\
&\overset{\text{חוקי}}{=} r_{t+1} \left(W - \sum_{i=1}^{n} y_i w_i \right) \ge 0
\end{aligned}$$

כרצוי. ■

1.1 תכנון לינארי

ניתן להסתכל על ווקטור הקלט $(x_1,\ldots,x_n) \in \mathbb{R}^n$ עם $0 \le x_i \le 1$ כנקודה בתוך קוביה n מימדית עם צלעות באורך 1. מכאן עלינו למצוא נקודה בתוך הקוביה שנותנת לנו פונקציית רווח מקסימלית $\sum_{i=1}^{n} x_i v_i$.

בבעיות הבאות נראה שלא ניתן להציג פתרון מלא, אלא בצורה איטרטיבית. נרצה אם כך לפתור איטרטיבית את בעיית התרמיל השברי.

1.2 פתרון איטרטיבי לבעיית התרמיל השברי

קלט $n =$ מספר הפריטים בחנות. $W =$משקל מירבי שהתרמיל יכול לשאת.$(v_1, w_1), \ldots, (v_n, w_n)$ כאשר v_i הוא הערך הכולל של הפריט ה-i ו-w_i המשקל הכולל של הפריט ה-i. הפריטים ניתנים לחלוקה.

פלט רשימה של מספרים $x_1, x_2, \ldots, x_n$ כאשר x_i היא חלקיות הפריט ה-i בתוך התרמיל, כך ש-$\sum_{i=1}^{n} x_i w_i \le W$ ו-$\sum_{i=1}^{n} x_i v_i$ מקסימלי.

Algorithm 1 בעיית התרמיל השברי

1 : Claculate the effective values of the elements in decending order. // $O(n \log n)$

2 : Initilization: $k = \emptyset$

3 : Iteration: pass through the elements in order // $O(n)$

4 : Put as much as possible from the i^{th} element.

5 : Until: the Knapsack is full and then we return k

Complexity: $O(n \log n)$

Space: $O(n)$

באופן כללי, בקורס נרצה לפתור בעיית אופטימיזציה. כל בעיה מגדירה את מרחב הפתרונות החוקיים $\mathcal{S}$ ופונקציית ערך $f:\mathcal{S}\to\mathbb{R}_+$. נרצה למצוא $\mathcal{S}^*\in\mathcal{S}$ כך ש-$f\left(\mathcal{S}^*\right)$ מקסימלית (מינימלית). אלגוריתם חמדני עובד בצורה כזו שבכל איטרציה נדע איך להגיע לצעד הבא כך שהפתרון ישאר אופטימלי. במילים אחרות אנו בונים מסלול של צעדים, ומוכיחים שהמסלול מקסימלי. בחירת הצעד נקראת "בחירה חמדנית".

כדי להוכיח שדרך כזו תביא אותנו לפתרון אופטימלי, נשתמש בעקרון של שיפורים מקומיים. לכומר, נראה שאם בשלב מסוים האלגוריתם אל עושה בחירה חמדנית הוא טועה (למשל ניתן להחליף את בחירתו בבחירה חמדנית ורק לשפרו).

הטענה אומרת שבכל שלב כדאי לאלגוריתם לבצע בחירה חמדנית, נקראת (בכל בעיה שנראה) "למת החלפה". למת ההחלפה מפרמלת את העקרון החמדני.

משפט. (למת ההחלפה לבעיית התרמיל השברי) יהי $y=(y_1,\ldots,y_n)$ פתרון חוקי לבעיה. נניח כי קיים אינדקס $1\le j\le n$ כך ש-$y_j<1$ וכך ש-$\sum_{i=1}^{j}y_iw_i<W$ אזי קיים פתרון חוקי $z=(z_1,\ldots,z_n)$ כך ש-$z_1=y_1,\ldots,z_{j-1}=y_{j-1}$ כך ש-$z_j>y_j$ וכך ש-$\sum_{i=1}^{j}z_iv_i>\sum_{i=1}^{j}y_iv_i$. במילים אחרות y אינו אופטימלי.

הוכחה: נניח בלי הגבלת הכלליות כי הערכים הסגוליים של כל הפריטים שונים זה מזה, כלומר $r_1>r_2>\ldots.>r_n$ כי אם שני פריטים עם אותו ערך סגולי נוכל לאחד אותם לפריט אחד ולקבל בעיה דומה וקלה יותר. נחלק למקרים.

אם $\sum_{i=1}^{n}y_iw_i<W$ אזי ניתן למלא את התרמיל עוד עם y_j גדול יותר.

אחרת, $\sum_{i=1}^{n}y_iw_i=W$. נטפל במקרה זה. מתקיים אם כן כי $\sum_{i=1}^{n}y_iw_i=W$ על פי הנחת הלמה, $\sum_{i=1}^{j}y_iw_i<W$. מכך קיים $k>j$ כך ש-$y_k>0$. נגדיר פתרון חדש z באופן הבא, $z_i=y_i$ לכל $i\ne j,k$. נגדיר $d_j=(1-y_j)\,w_j$ המשקל של הפריט ה-j מחוץ לתרמיל. נגדיר $d_k=y_kw_k$ להיות המשקל של הפריט בתוך התרמיל. נגדיר $0<d=\min\{d_k,d_j\}$ שכן $y_j<1\Rightarrow d_j>0$. נגדיר $z_k=y_k-\frac{d}{w_k},z_j=y_j+\frac{d}{w_j}$. נראה כי z פתרון חוקי. לכל $i\notin\{j,k\}$ מתקיים כי $0\le z_i=y_i\le1$. נראה כי $z_j,z_k\le1$. מתקיים

$$0\le y_j<z_j=y_j+\frac{d}{w_j}\le y_j+\frac{d_j}{w_j}=y_j+\frac{(1-y_j)\,w_j}{w_j}=y_j+1-y_j=1$$

וכן

$$0=y_k-\frac{y_kw_k}{w_k}\le z_k=y_k-\frac{d}{w_k}<y_k\le1$$

כרצוי.

עתה, נראה כי $\sum_{i=1}^{n}z_iw_i=\sum_{i=1}^{n}y_iw_i$. נחשב

$$\begin{aligned}\sum_{i=1}^{n}z_iw_i-\sum_{i=1}^{n}y_iw_i&=\sum_{i=1}^{n}(z_i-y_i)\,w_i=(z_j-y_j)\,w_j+(z_k-y_k)\,w_k\\&=\frac{d}{w_j}w_j-\frac{d}{w_k}w_k=d-d=0\end{aligned}$$

לכן הפתרון חוקי. נראה כי $\sum_{i=1}^{n}z_iv_i>\sum_{i=1}^{n}y_iv_i$, נחשב

$$\begin{aligned}\sum_{i=1}^{n}z_iv_i-\sum_{i=1}^{n}y_iv_i&=\sum_{i=1}^{n}(z_i-y_i)\,v_i=(z_j-y_j)\,v_j+(z_k-y_k)\,v_k\\&=\frac{d}{w_j}v_j-\frac{d}{w_k}v_k=d\cdot r_j-d\cdot r_k\\&=d\cdot(r_j-r_k)>0\end{aligned}$$

ולכן הפתרון החדש טוב יותר. ■

משפט. (אופיטמילות האלגוריתם) האלגוריתם החמדן מחזיר פתרון אופיטמלי לבעיה.

הוכחה: נראה כי יש פתרון אופטימלי.

נשים לב כי מרחב הפתרונות החוקיים לבעיה זו הוא $\mathcal{S} = \left\{(x_1, \ldots, x_n) \in \mathbb{R}^n : 0 \leq x_i \leq 1, \sum\limits_{i=1}^{n} x_i w_i \leq W\right\}$ קבוצה סגורה וחסומה ב-$\mathbb{R}^n$ ולכן היא קבוצה קומפקטית. פונקציית הערך $f(x) = \sum\limits_{i=1}^{n} x_i v_i$ היא פונקציה רציפה על $\mathcal{S}$ ולכן מקבלת מקסימום על $\mathcal{S}$, ממשפט וירשטראס, לכן לבעיה זו יש פתרון אופטימלי.

נראה כי כל פתרון חוקי y השונה מהפתרון החמדן מקיים את תנאי למת ההחלפה, ולכן על פי למת ההחלפה אינו אופטימלי. נשים לב כי הפתרון החמדן נראה כך,

יהי $0 \leq t \leq n-1$ כך ש-$\sum\limits_{i=1}^{t} w_i \leq W$ והסכום $\sum\limits_{i=1}^{t+1} w_i > W$ אזי $x_1 = \ldots = x_t = 1$ ו-$x_{t+1} = \frac{W - \sum\limits_{i=1}^{t} w_i}{w_{t+1}}$ וגם $x_{t+2} = \ldots = x_n = 0$.

נבחין כי $\sum\limits_{i=1}^{t+1} x_i w_i = W$. יהי $y = (y_1, y_2, \ldots, y_n)$ פתרון חוקי השונה מ-x. יהי $1 \leq j \leq n$ האינדקס הראשון כך ש-$y_j \neq x_j$, נראה כי $x_j < y_j$. נניח בשלילה כי $x_j > y_j$, נזכור כי $x_1 = \ldots = x_t = 1$ ולכן בהכרח $j \geq t+1$. לכן

$$\sum_{i=1}^{j} y_i w_i > \sum_{i=1}^{j} x_i w_i \geq \sum_{i=1}^{t+1} x_i w_i = W$$

בסתירה לכך ש-y פתרון חוקי.

לכן הוכחנו כי $x_j < y_j$ ולכן $x_j < y_j \leq 1$ וגם מתקיים

$$\sum_{i=1}^{j} y_i w_i < \sum_{i=1}^{j} x_i w_i \leq W$$

לכן y מקיים את תנאי למת ההחלפה ולכן אינו אופטימלי. ■

2 בעיית שיבוץ משימות

2.0.0.1 "סיפור מסגרת" נתונות n משימות, כך שלא ניתן לבצע שתי משימות במקביל. המטרה לבחור מספר כמה שיותר גדול של משימות שכן ניתן לבצע אותן.

קלט n קטעים סדורים על ציר הזמן. הקטעים ניתנים על ידי נקודת ההתחלה ונקודת הסיום שלהם $[s_1, f_1], [s_2, f_2], \ldots, [s_n, f_n]$.

פלט תת קבוצה $S \subseteq [n]$ כך שהקטעים עם אינדקסים ב-S זרים זה לזה וכך ש-$|S|$ מקסימלי.

דוגמה. $n = 5$, עם הקטעים

$$\begin{array}{ccccc} & & & & - \\ & & & - & \\ & - & & & \\ - & - & & & \\ - & - & - & - & - \end{array}$$

במקרה זה הפתרון האופטימלי הוא 3, למשל $S = \{3, 4, 5\}$ או $S = \{2, 4, 5\}$.

2.1 האלגוריתם

מרחב הפתרונות החוקיים הוא {הקטעים שהאינדקסים מייצגים זרים $\mathcal{S} = \{S \subseteq [n]\,,$. על המרחב $\mathcal{S}$ מוגדרת פונקצית ערך $q : \mathcal{S} \to \mathbb{R}_+$ עם $q(S) = |S|$. המטרה שלנו היא למצוא נקודת מקסימום של q על $\mathcal{S}$.

2.1.0.1 פתרון נאיבי נעבור על כל $\mathcal{S}$ וכך נמצא את נקודות המקסימום של q. זמן הריצה יהיה $\Omega(|\mathcal{S}|) \geq \Omega(2^n)$.

נחפש אלגוריתם חמדני איטרטיבי הפותר את הבעיה באופן הבא. בכל שלב, האלגוריתם בוחר קטע נוסף, על פי עקרון חדמני ומוסיף אותו לפתרון.

בעיה. איזה עקרון חמדני נבחר?

נציע כמה עקרונות:

1. בכל שלב נבחר קטע המתנגש עם המספר הקטן ביותר של קטעים שנשארו וכמובן לא חותך את הקטעים שכבר בחרנו.
2. נוסיף את המשימה הקצרה ביותר.
3. הקטע שמתחיל כמה שיותר מהר מהמשימה האחרונה שהוספנו.
4. נוסיף את המשימה שזמן הסיום שלה מינימלי ואיננה מתנגשת עם הקטעים שבחרנו עד כה.

מסתבר שהפתרון הנכון הוא עם תנאי 4. מוזר, אבל נראה שלכל אחד משלוש ההצעות האחרות יש דוגמאות נגדיות.

דוגמאות

1. נבחר את הקטעים הבאים:

```
1  –                           –  5
2  –  –                     –  –  6
3  –  –  –               –  –  –  7
4  –  –  –  –         –  –  –  –  8
   9  –  –  –  |   –  –  –    10
            –  –  –
               11
```

אנו נקבל את הפתרון $\{11, 1, 5\}$ אבל הפתרון האופטימלי הוא $\{1, 5, 9, 10\}$.

2. נבחר את הקטעים הבאים:

```
      1
   –  –  –          2
            –  –  –
      –  –  –
         3
```

נקבל 1 אבל הפתרון האופטימלי הוא 2.

3. נבחר את הקטעים הבאים

$$\begin{array}{ccccc} 1 & & & & \\ - & - & - & - & - \\ & - & & & \\ & 2 & & - & \\ & & & 3 & \end{array}$$

אזי הפתרון שנקבל הוא 1 אבל הפתרון האופטימלי הוא 2.

אלגוריתם 2 בעיית שיבוץ המשימות

האלגוריתם יפעל באופן הבא:

- PreProccessing: נמיין את הקטעים בסדר עולה לפי זמני הסיום $f_1 \leq f_2 \leq \ldots \leq f_n$.
- <u>אתחול</u> $A = [n]$ מכיל את האינדקסים של המשימות שעדיין ניתן לבחור.
  - ◁ נאתחל $G = \emptyset$
- <u>איטרציה</u> בכל שלב נעביר מ-A ל-G את האינדקס המינימלי ב-A.
  - ◁ נמחק מ-A את הקטעים החותכים בקטע שכרגע בחרנו.
- <u>סיום</u> כאשר $A = \emptyset$ נעצור ונחזיר את G.

2.2 זמן ריצה

השלב הראשון של המיון עולה $\Theta(n \log n)$, האתחול והסיום $\Theta(1)$, ניתן לממש זאת באמצעות מבנה נתונים מספיק יעיל, וכמות האיטרציות היא $O(n)$. לכן סך הכל הסיבוכיות היא $\Theta(n \log n)$.

2.3 נכונות האלגוריתם

נסמן ב-$G = (g_1, g_2, \ldots, g_k)$ אינדקסי הקטעים שבחרנו, כאשר g_{i+1} הוא הקטע בעל זמן הסיום הקצר ביותר אחרי g_i כך שהוא לא חותך את g_i.

יהי פתרון חוקי $S = (s_1, s_2, \ldots, s_m)$, נוכיח כי $k \geq m$.

רעיון ההוכחה הוא החלפה של g_i עם s_i עד שנבצע $\min\{k, m\}$ החלפות, לאחר מכן נוכל לטעון כי את כל שאר איברי S אפשר להכניס לפתרון אבל זה בסתירה לכך שהקבוצה A ריקה. נסיים בשיעור הבא.

למעשה האיטואיציה החמדנית היא שאם איננו מתנגשים אם כל הקטעים הבאים, כל הקטעים שנגמרים לפנינו אינם מתנגשים איתם.

ננסח את הלמה הבאה:

למה. יהי $S = \{i_1, \ldots, i_m\}$ פתרון חוקי לבעיה ויהי $1 \leq j \leq m$ האינדקס הראשון כך ש-$g_j \neq i_j$. אזי גם הפתרון $(g_1, \ldots, g_j, i_{j+1}, \ldots, i_m)$ הינו פתרון חוקי.

יחד עם זאת, אפשר לנסח אותה באופן שונה.

למה. יהי $S = \{i_1, \ldots, i_m\}$ פתרון חוקי לבעיה כאשר $i_1 < i_2 < \ldots < i_m$. יהי $1 \leq j \leq m-1$ כלשהו ויהי t האינדקס המינימלי של הקטע שלא חותך את הקטעים $i_1, \ldots, i_j$ אזי גם $S_1 = \{i_1, i_2, \ldots, i_j, t, i_{j+2}, \ldots, i_m\}$ הוא פתרון חוקי.

הוכחה: עלינו להראות שהקטעים ב-S_1 זרים זה לזה. מכיוון ש-S הוא פתרון חוקי, הקטעים $i_1, \ldots, i_j, i_{j+2}, \ldots, i_m$ זרים זה לזה. בנוסף, לפי בחירת t, הקטע t זר לקטעים $i_1, i_2, \ldots, i_j$, נוודא כי לכל $j+2 \leq k \leq m$ הקטעים t, i_k זרים זה לזה.

<u>אבחנה מרכזית:</u> יהיו $[s,f],[s',f']$ שני קטעים זרים כך ש-$f'>f$ אזי $f<s'$, שכן אחרת, החיתוך בין הקטעים היה $\min\{f-s',f-s\}>0$ בסתירה לכך שהם זרים, או בפשטות, הנקודה f הייתה בשתי הקטעים בסתירה להנחה.

נשים לב ש-i_k ו-i_{j+1} שייכים לפתרון חוקי S ולכן שניהם זרים. בנוסף, מהזרות, $i_{j+1}<i_k$ ולכן $f_k>f_{j+1}$. לפי האבחנה, זה גורר כי $S_{i_k}>f_{i_{j+1}}$. בנוסף, הקטע i_{j+1} זר לקטעים $i_1,\ldots,i_j$ כי כל הקטעים האלה נמצאים באותו פתרון חוקי S. נזכור כי הקטע ה-t נבחר כקטע בעל אינדקס מינימלי שלא חותך את הקטעים $i_1,i_2,\ldots,i_j$ ולכן מתקיים $t\le i_{j+1}$ ולכן $f_t\le f_{i_{j+1}}<S_{i_k}$ והקטעים t,i_k זרים זה לזה. ■ נרצה עתה להוכיח את אופטימליות הפתרון החמדני. ניתן קצת אינטואיציה.

נתחיל מפתרון $S^*=(i_1,\ldots,i_m)$. בעזרת למת ההחלפה, נתחיל לשנות את הפתרון S^* עד שנגיע ל-G, כלומר נבנה מסילה במרחב הפתרונות עד שנגיע לנקודת האופטימום. מהלמה ניתן להחליף את i_1 ב-g_1. אבל מהלמה ניתן להחליף גם את i_2 בקטע המינימלי שלא חותך את g_1, שהוא g_2. כך נמשיך עד שנחליף את i_r ב-g_r האחרון. מכאן נקבל פתרון $S_r^*:g_1,\ldots,g_r,i_{r+1},\ldots,i_m$. נוכיח כי $r\ge m$ ונקבל ש-G אופטימלי. מדוע לא יתכן אחרת? אם יש קטע זר ל-$g_1,\ldots,g_r$, אז האלגוריתם החמדן לא עצר, כי עצרנו כש-A ריקה.

אנחנו נוכיח זאת בדרך חדשה. לכל פתרון אופטימלי, נגדיר פונקציה שתתן את הרישא המשותפת עם הפתרון החמדן. אנו ניקח פתרון עם רישא מקסימלית ונוכיח שגדול הרישא שווה ל-r.

משפט. הפתרון החמדן הינו פתרון אופטימלי לבעיית שיבוץ המשימות.

הוכחה: נוכיח חוקיות

<u>חוקיות הפתרון החמדן:</u> נשים לב שלפי דרך פעולתו של האלגוריתם החמדני, לפני תחילת כל איטרציה הקבוצה A מכילה רק קטעים שזרים לכל הקטעים בקבוצה G. לכן בסוף האיטרציה הקבוצה G מכילה קטעים זרים ולכן היא פתרון חוקי. זה המצב גם בסוף הריצה של האלגוריתם ולכן הפתרון G המוחזר על ידי האלגוריתם הינו פתרון חוקי.

יהי $G=(g_1,g_2,\ldots,g_r)$ הפתרון החמדן כאשר $g_1<g_2<\ldots<g_r$. עבור פתרון חוקי לבעיה $S=(i_1,i_2,\ldots,i_k)$, נגדיר את גודל הרישא המקסימלית של S ו-G לפי המקרים הבאים:

(i): $k\ge r$. מתקיים כי $i_1=g_1,i_2=g_2,\ldots,i_r=g_r$. במקרה זה נאמר כי לשני הפתרונות רישא משותפת בגודל r.

(ii): קיים $0\le l\le r$ כך שמתקיים כי $i_1=g_1,i_2=g_2,\ldots,i_l=g_l$ אבל $i_{l+1}\ne g_{l+1}$. במקרה זה נאמר כי לשני הפתרונות רישא משותפת בגודל l.

יהי $S^\star$ פתרון אופטימלי שלו רישא משותפת מקסימלית מבין כל הפתרונות האופטימליים עם G. נראה כי $S^\star=G$ ובכך נוכיח כי G אופטימלית. נוכיח שני שלבים, כי הרישא בגודל r וכי $S^\star=G$.

<u>שלב ראשון:</u> נראה כי הרישא המשותפת של $S^\star=(i_1,i_2,\ldots,i_m)$ ו-G היא בגודל r. נניח שלא, אזי קיים $l<r$ כך ש-$i_1=g_1,\ldots,i_l=g_l$ אבל $i_{l+1}\ne g_{l+1}$. לפי למת ההחלפה, ניתן להחליף את הבחירה ה-$l+1$ ב-t כאשר t הוא האינדקס המינימלי של קטע שלא חותך את $g_1,\ldots,g_l$. לפי דרך פעולותו של האלגוריתם החמדן, $t=g_{l+1}$ ולכן קיבלנו פתרון חוקי חדש $S_1=(g_1,\ldots,g_l,g_{l+1},i_{l+2},\ldots,i_m)$. מכאן ש-$m=|S^\star|=|S_1|$ הוא פתרון אופטימלי עם רישא משותפת עם G הגדולה מזו של $S^\star$ בסתירה להגדרת $S^\star$.

<u>שלב שני:</u> נוכיח עתה כי $S^\star=G$. הראינו בשלב הראשון כי $G\le S^\star$. נניח בשלילה כי קיים קטע ב-$S^\star$ שלא נמצא ב-G, אזי $i_{r+1}\ne G$. נשים לב כי על פי דרך פעולתו של האלגוריתם החמדן, הקטע i_{r+1} שייך לקבוצה A בסוף כל איטרציה של האלגוריתם. לכן $i_{r+1}\in A$ גם בסוף הריצה של האלגוריתם החמדני, ולכן הוא עוצר כאשר $A\ne\emptyset$ בסתירה לדרך פעולתו. ■

3 וקטורים בלתי תלויים לינארית

3.1 מבוא

בעיה. נרצה למצאו קבוצת וקטורים בלתי תלויים לינארית בעלת משקל מקסימלי.

קלט קבוצה סופית $X=\{v_1,v_2,\ldots,v_n\}$ במרחב וקטורי t-מימדי שנסמנו V, וגם פונקציית משקל $\mu:X\to\mathbb{R}$.

פלט תת קבוצה $S\subseteq[n]$ כך שהווקטורים עם אינדקסים ב-S בת"ל וכך ש-$\mu(S)$ מקסימלי, כאשר $\mu(S)=\sum\limits_{i\in S}\mu(v_i)$.

הערה. מכאן נסיק כי וקטורים בעלי משקל אפס, אינם מעניינים.

דוגמה. $n = 5$ ו-$V = \mathbb{R}^4$ עם הווקטורים
$$X = \begin{pmatrix} v_1 & v_2 & v_3 & v_4 & v_5 \\ 1 & 1 & 1 & 3 & e \\ 2 & -1 & \sqrt{2} & 2+2\sqrt{2} & \pi \\ 3 & 1 & \sqrt{3} & 3+2\sqrt{3} & -\sqrt{e} \\ 4 & -1 & 2 & 8 & -\sqrt{\pi} \end{pmatrix}$$
עם המשקלים $(5,3,7,4,2)$ בהתאמה. נתון כי $v_4 = v_1 + 2v_3$ זו התלות היחידה. הפתרון האופטימלי הוא $S_1^\star = \{3,1,2,5\}$.

בעיה. מה אם כך הוא הפתרון החמדני?

הצעה. נבחר את הוקטור בעל המשקל המקסימלי, שיחד עם הווקטורים שלקחנו עד כה איננו יוצר תלות.

שאלה כיצד זה שונה מהאלגוריתם קרוסקל?

תשובה בעיה זו היא הכללה של האלגוריתם קרוסקל, ולמעשה ניתן להתאים לגרף ווקטורים בצורה כזו שסגירת מעגל תהווה יצירת תלות לינארית. נראה זאת בהרחבה בקרוב.

נציג אם כך אלגוריתם חמדני לפתרון הבעיה:

אלגוריתם 3 וקטורים בלתי תלויים לינארית בעלי משקל מקסימלי

האלגוריתם.

- עיבוד מידע מוקדם: נמיין את הווקטורים על פי משקלם בסדר יורד.

  ◁ נניח מעתה כי $\mu(v_1) \geq \mu(v_2) \geq \ldots \geq \mu(v_n) > 0$.

- אתחול: $A = [n]$ וגם $G = \emptyset$.
- איטרציה: נעבור על האיברים ב-A.

  ◁ בכל איטרציה, נעביר מ-A ל-G את האינדקס המינימלי ב-A.

  ◁ נמחק מ-A את כל הווקטורים התלויים לינארית בווקטורים ב-G. זאת לצורך חוקיות.

- כאשר $A = \emptyset$ נעצור ונחזיר את G.

3.2 זמן ריצה

זמן הריצה של השלב הראשון הוא $\Theta(n \log n)$ מיון הווקטורים. השלב השני והשלישי $\Theta(1)$ ושלב האיטרציה הוא n כפול הזמן שלוקח להחליט האם וקטור תלוי לינארית או לא בווקטורים שיש לנו ב-G. זו בעיה אלגוריתמית בפני עצמה,
נרצה לדעת האם עבור $u_1, \ldots, u_k$ קיימים מקדמים $x_1, \ldots, x_k$ כך ש-$u = \sum_{i=1}^{n} x_1 v_i$ או באופן שקול כך ש- $\underbrace{\begin{bmatrix} | & | & & | \\ u_1 & u_2 & \ldots & u_k \\ | & | & & | \end{bmatrix}}_{\in \mathbb{M}_{t \times k}} \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} = \begin{pmatrix} | \\ w \\ | \end{pmatrix}$. את מערכת זו ניתן לפתור באמצעות אלגוריתם הדירוג של גאוס, בכל שלב נבצע פעולת שורה שעולה לנו $t \cdot k$ שכן אנו מחסירים שורה אחת מ-t שורות וכל החסרה עולה k, אנו מבצעים זאת עבור k עמודות, ולכן נקבל $t \cdot k^2$ נבחין כי $1 \leq k \leq n$ ולכן נוכל לחסום עלות זו על ידי $t \cdot n^2$. בנוסף, בדיקת קיום הפתרון למערכת מתבצעת עבור לכל היותר n ווקטורים ב-G ולכן נקבל $t \cdot n^3$ סך הכל. כלומר זמן הריצה של האלגוריתם הוא $\Theta(t \cdot n^3)$ סך הכל. אם $t = 2^n$ האלגוריתם עדיין יעיל, שכן אם t עצום, הקלט עצום וקיבלנו למעשה פתרון שהוא $m \cdot \log^3 m$ בגודל הקלט כאשר $m = 2^n$.

3.3 הוכחת נכונות

את אלגוריתם זה נוכיח באמצעות למה שונה, שתכליל בעיות מסוג זה (נראה בהרחבה בקרוב), והיא "למת החלפה מטרואידית" שאיננה מסתכלת על פונקציית משקל ספציפית, אלא על המבנה הקומבינטורי של מרחב הפתרונות.

למה. יהי V מרחב וקטורי ותהיינה X,Y שתי קבוצות סופיות של וקטורים בת"ל ב-V כך ש-$|X|<|Y|$ אזי קיים $y\in Y\backslash X$ כך ש-$X\cup\{y\}$ בת"ל.

הערה. (תזכורת מאלגברה לינארית) יהי U מרחב וקטורי סוף מימדי. בסיס של U זו קבוצה בת"ל של וקטורים ב-U הפורשת את U. כל הבסיסים של U הם באותו הגודל הנקרא המימד של U. U לא יכול להכיל קבוצה בת"ל שגדולה יותר מהמימד שלו.

הוכחה: (למת ההחלפה) יהי U תת המרחב של V הנפרש על ידי X. אזי מהיות X בת"ל נקבל כי $\dim(U)=|X|$. מכיוון ש-$|X|<|Y|$ ו-Y בת"ל מתקיים כי $U\not\supseteq Y$ כלומר קיים $y\in Y$ כך ש-$y\notin U$. מכאן y איננו צירוף לינארי של וקטורים מ-X. לכן $y\notin X$ ו-$X\cup\{y\}$ בת"ל. ■

משפט. האלגוריתם החמדני מחזיר פתרון אופטימלי לבעיה.

הוכחה: יהי G הפתרון החמדן. יהי $S^\star$ פתרון אופטימלי.

שלב ראשון: נראה כי $|G|=|S^\star|$. נניח שלא,
אזי אם $|G|<|S^\star|$ אזי על פי למת ההחלפה, קיים וקטור $s\in S^\star$ כך ש-$s\notin G$ וגם $G\cup\{s\}$ בת"ל. לכן הווקטור s אינו תלוי לינארית ב-G ולכן על פי דרך פעולתו של האלגוריתם החמדן, הווקטור נשאר בקבוצה A לאחר כל איטרציה של האלגוריתם ולכן גם בסיום הריצה של האלגוריתם. לכן האלגוריתם החמדני עצר כאשר $A\neq\emptyset$ שכן היא מכילה את s, בסתירה לדרך פעולת האלגוריתם.
אחרת, אם $|G|>|S^\star|$, אזי לפי למת ההחלפה קיים $g\in G\backslash S^\star$ כך ש-$S^\star\cup\{g\}$ בת"ל ולכן $S_1=S^\star\cup\{g\}$ פתרון חוקי שמשקלו $\mu(S_1)=\mu(S^\star)+\mu(g)>\mu(S^\star)$ בסתירה לאופטימליות $S^\star$.

שלב שני: נוכיח כי $\mu(G)=\mu(S^\star)$. אינטואיטיבית, אם בשלילה $\sum\limits_{i=1}^{k}\mu(g_i)=\mu(G)<\mu(S^\star)=\sum\limits_{i=1}^{k}\mu(v_i)$ לכן קיים $1\le j\le k$ כך ש-$\mu(g_j)<\mu(v_j)$. נרשום את שני הפתרונות

$$
\begin{array}{ll}
S^\star & \underbrace{v_1,v_2,\ldots,v_j}_{Y},\ldots,v_k \\[2ex]
G & \underbrace{g_1,g_2,\ldots,g_{j-1}}_{X},g_j,\ldots,g_k
\end{array}
$$

אבל מכאן מקבלים כי $v_1,v_2,\ldots,v_j$ בעלי משקל גדול מ-g_j, שכן הם מסודרים בסדר יורד לפי משקל. לכן מלמת ההחלפה, ניתן לבחור ווקטור אחד מביניהם שבלתי תלוי ב-$g_1,\ldots,g_{j-1}$ ולכן האלגוריתם שלנו יכול היה לבחור בחירה טובה יותר מ-g_j וזו סתירה לחמדנותו. עתה נוכיח זאת פורמלית.
נניח בשלילה כי $\mu(G)\neq\mu(S^\star)$ אזי $\mu(G)<\mu(S^\star)$ מאופטימליות $S^\star$. נסמן $G=\{g_1,\ldots,g_k\}$ וגם $S^\star=\{i_1,\ldots,i_k\}$ כאשר הערכים בקבוצות מסודרים בסדר עולה. על פי ההנחה,

$$\mu(S^\star)=\sum_{j=1}^{k}\mu(i_j)>\sum_{j=1}^{k}\mu(g_j)=\mu(G)$$

ולכן קיים $1\le j\le k$ כך ש-$\mu(g_j)<\mu(i_j)$. נסמן $X=\{g_1,g_2,\ldots,g_{j-1}\}$ ו-$Y=\{i_1,i_2,\ldots,i_j\}$ קבוצות בת"ל כתת קבוצות של הפתרונות חוקיים. נבחין כי $|X|<|Y|$ ולכן לפי למת ההחלפה, קיים $y\in Y\backslash X$ עבורו $X\cup\{y\}$ בת"ל. באופן יותר מפורש, קיים $1\le t\le j$ כך ש-$\{g_1,g_2,\ldots,g_{j-1}\}\cup\{i_t\}$ בת"ל. לכן הווקטור i_t אינו תלוי לינארית ב-$g_1,g_2,\ldots,g_{j-1}$ ולכן הוא היה ניתן לבחירה באיטרציה ה-j של האלגוריתם החמדן. בנוסף, מכיוון ש-$t\le j$ מתקיים כי $i_t\le i_j$ ולכן $\mu(g_j)<\mu(i_j)\le\mu(i_t)$, יוצא שהאלגוריתם החמדן לא בחר בחירה חמדנית באיטרציה ה-j, בסתירה לדרך פעולתו. ■

4 מטרואידים

ראינו הרבה בעיות אלגוריתמיות שפתרנו בדרכים חמדניות, והיינו רוצים להכליל אותן לכדי בעיה כללית שממנה נסיק פתרון לבעיות שראינו. מטרואיד הוא כלי לביצוע משימה זו. יחד עם זאת, קשה למצוא דוגמא פשוטה למטרואיד ודוגמאות מסובכות יש לא מעט.
למעשה, רוב הבעיות האלגוריתמיות שראינו עד עכשיו, מלבד בעיית התרמיל השברי, היה אפשר לכתוב באופן הבא:

בעיה. בעיה אלגוריתמית גנרית.

קלט קבוצה סופית של אובייקטים $B=\{x_1,x_2,\ldots,x_n\}$, פונקציית משקל $\mu:B\to\mathbb{R}$ ומשפחה I של תת קבוצות של B המגדירה את אוסף הפתרונות החוקיים.

פלט תת קבוצה $S\subseteq B$ כך ש-$S\in I$ וכך ש-$\mu(S)$ מקסימלי כאשר $\mu(S)=\sum\limits_{x_i\in S}\mu(x_i)$.

באמצעות פתרון לבעיה כללית זו, נוכל להסיק פתרון לבעיות רבות.

דוגמאות

1. קבוצת וקטורים בת"ל עם משקל מירבי. במקרה זה $B=\{v_1,v_2,\ldots,v_n\}$, $I=\{$כל תתי הקבוצות שהן בת"לב-B$\}$ ופונקציית המשקל μ היא כפי שהיא מוגדרת בדוגמא.
2. בעיית שיבוץ משימות. במקרה זה $B=\{[s_1,f_1],\ldots,[s_n,f_n]\}$ וגם $I=\{S\subseteq[n]\mid$ הקטעים עם האינדקסים ב-S זרים זה לזה$\}$. פונקציית המשקל $\mu:[n]\to\mathbb{R}$ מוגדרת על ידי $\mu(i)=1$ לכל $1\le i\le n$.
3. גרסא שנייה. במקום לחפש תת קבוצה של משימות לא מתנגשות בגודל מקסימלי, נחפש תת קבוצה הממקסמת את הזמן הכולל של המשימות שגם לא מתנגשות. זה שייך גם לאותה מסגרת עם $\mu(i)=f_i-s_i$ ו-B,I זהות.
4. התרמיל השלם. בשונה מבעיית התרמיל השברי, כן הפריטים לא ניתנים לחלוקה.

4.1 בעיית התרמיל השלם

קלט $x_1=(v_1,w_1),\ldots,x_n=(v_n,w_n)$ משקל וערך. W המשקל המירבי של התרמיל. $\mu(x_i)=v_i$ ו-$I=\left\{S\subseteq[n]\mid\sum\limits_{i\in S}w_i\le W\right\}$.

פלט קבוצה $\{i_1,\ldots,i_m\}\in I$ עם משקל מקסימלי.

נציע את האלגוריתם הבא שמשתמש בווקטורים בלתי תלויים מקסימלים:

אלגוריתם 4 בעיית התרמיל השלם, אלגוריתם גנרי

- עיבוד מידע מוקדם: נמיין את הפריטים על פי משקלים בסדר יורד $\mu(x_1)\ge\mu(x_2)\ge\ldots\ge\mu(x_n)$.
- אתחול: $A=[n]$ ו-$G=\emptyset$.
- איטרציה: בכל שלב נעביר מ-A ל-G את האינדקס הנמוך ביותר ונמחק מ-A את כל האינדקסים x כך ש-$G\cup\{x\}\notin I$.
- סיום: כאשר $A=\emptyset$ נעצור ונחזיר את G.

ננתח את זמן הריצה:

- $\Theta(n\log n)$ למיון המשקלים.
- T - זמן ריצה לפתרון של הבעיה האלגוריתמית הבאה: בהנתן תת קבוצה $S\subseteq B$ האם $S\in I$?

◁ אנו חוזרים על שלב זה n פעמים ולכן $\Theta\left(n\cdot T\right)$.

- סך הכל $\Theta\left(n\left(\log n+T\right)\right)$.

שאלה האם זה אלגוריתם אופטימלי?

שאלה מתי האלגוריתם החמדן הגנרי מחזיר תשובה אופטימלית לבעיה?

משפט. נתבונן בבעיה הגנרית. אם הזוג $\left(B,I\right)$ הוא מטרואיד, אזי האלגוריתם החמדן הגנרי מחזיר פתרון אופטימלי לבעיה לכל פונקציית משקל.

הערה. נבחין כי האלגוריתם משתמש בתורשתיות של המטרואיד, שכן בכל פעם אנו בודקים האם אנו עדיין בתוך I.

הוכחה: (המשפט) תרגיל: בדיוק כמו בבעיית קבוצת וקטורים בת"ל עם משקל מקסימלי. ■

משפט. (אי נכונות האלגוריתם הגנרי ללא החלפה) אם המשפחה I של תת קבוצות של B הינה לא ריקה ותורשתית, אבל לא מקיימת את תכונת ההחלפה אזי קיימת פונקציית משקל $\mu:B\to\mathbb{R}_{+}$ כך שהאלגוריתם החמדן הגנרי לא מחזיר פתרון אופטימלי לבעיה המתאימה.

מסקנה. ממשפט נכונות האלגוריתם הגנרי, ניתן להסיק שהמשפחות I בבעיית שיבוץ משימות ובבעיית התרמיל השלם הן (כנראה) אינן מטרואידים.

הוכחה: נמצא דוגמא נגדית עבור שיבוץ משימות: $\begin{array}{c} |\quad -\quad |\quad |\quad -| \\ |\quad -\quad -\quad | \end{array}$. ■

הוכחה: (אי נכונות האלגוריתם הגנרי ללא החלפה) לפי הנתון קיימות קבוצות $S,T\in I$ כך ש-$|S|<|T|$ אבל לכל $t\in T\backslash S$ מתקיים $S\cup\left\{t\right\}\notin I$. נבנה פנקציית משקל μ באופן הבא:

$$\mu\left(x\right)=\begin{cases} 1 & x\in S\\ 1-\varepsilon & x\in T\backslash S\\ \varepsilon & x\notin S\cup T \end{cases}$$

כאשר $\varepsilon=\frac{|T|-|S|}{2|B|}$.

נרצה להראות שהאלגוריתם הגנרי המתאים ל-$\left(B,I\right)$ לא מחזיר פתרון אופטימלי. האינטואיציה היא שמתורשתיות האלגוריתם יבחר את האיברים ב-S במשקל 1, ויזרוק את האיברים ב-$T\backslash S$ ולכן הוא יקח לאחר מכן איברים מסביב, ששוקלים מעט מדי.

לפי דרך פעולתו של האלגוריתם החמדן הגנרי, מכיוון ש-$S\in I$ ו-I היא משפחה תורשתית, ב-$|S|$ האיטרציות הראשונות שלו, האלגוריתם החמדן יכניס ל-G את כל איברי הקבוצה S, שכן להם משקל מקסימלי. בשלב העדכון של האיטרציה מספר $|S|$, האלגוריתם ימחק מ-A את כל האיברים ב-$T\backslash S$ כי לכל $t\in T\backslash S$, מתקיים $S\cup\left\{t\right\}\notin I$. לכן הפתרון הסופי שנקבל G יקיים

$$\begin{aligned}\mu\left(G\right) & \leq\mu\left(S\right)+\mu\left(\left(S\cup T\right)^{c}\right)=|S|+\varepsilon\cdot|\left(S\cup T\right)^{c}|\\ & \overset{\star}{<}|S|+\varepsilon\cdot|B|=|S|+\frac{|T|-|S|}{2\left|B\right|}\left|B\right|\\ & =|S|+\frac{|T|-|S|}{2}=\frac{|S|+|T|}{2}\end{aligned}$$

$\star$: מתקיים כי $\left(S\cup T\right)^{c}\neq B$ כי $S\cup T\neq\emptyset$.

מצד שני,

$$\begin{aligned}\mu(T) &\geq (1-\varepsilon)\,|T| = |T| - \varepsilon\,|T| \geq |T| - \varepsilon\,|B| \\ &= |T| - \frac{|T|-|S|}{2\,|B|}\cdot|B| = \frac{|S|+|T|}{2} > \mu(G)\end{aligned}$$

ולכן קיים פתרון חוקי לבעיה (הקבוצה T) שמשקלו גדול יותר מזה של הפתרון החמדן, כלומר האלגוריתם המדן הגנרי לא מחזיר פתרון אופטימלי. ■

5 תרגול 2 - הוכחת נכונות של אלגוריתמים חמדניים

5.1 בעיית תא הדלק הקטן

קלט $N \in \mathbb{N}$ מספר הקילומטרים שניתן לעבור עם תא דלק מלא
$a_1, \ldots, a_n$ תחנות דלק במסלול בהן אפשר למלא את התא, כאשר $0 = a_1 < a_2 < \ldots < a_n$ וגם $a_i - a_{i-1} \leq N$ לכל $i > 1$.

פלט קבוצה $B = \{b_1, \ldots, b_m\}$ כך ש-$b_i < b_{i+1}$ לכל $1 \leq i \leq m-1$ וגם $b_{i+1} - b_i \leq N$ כך ש-$|B|$ מינימלי וגם $b_1 = a_1, b_m = a_m$.

אלגוריתם 5 אלגוריתם חמדן לפתרון בעיית תא הדלק הקטן

- <u>אתחול</u>: $B = \{a_1\}$.
- <u>איטרציה</u>: נעבור על כל התחנות הבאות לפי הסדר. נוסיף את a_i אם"ם אחד מהשניים מתקיים
  - ◁ $a_i = a_n$.
  - ◁ אין לנו מספיק דלק כדי להגיע לתחנה ה-$i+1$.
- <u>עצירה</u>: נחזיר את B.

כדי להוכיח את נכונות האלגוריתם נראה כי לכל פתרון בעל רישא כנ"ל, קיים פתרון אופטימלי המכיל אותו.

למה. יהי $B = \{b_1, \ldots, b_m\}$ אזי לכל $1 \leq k \leq m$ קיימים $c_{k+1}, \ldots, c_{m'}$ כך ש-$C = \{b_1, \ldots, b_k, c_{k+1}, \ldots, c_{m'}\}$ פתרון אופטימלי.

מסקנה. האלגוריתם החמדן שהצענו הוא אופטימלי.

הוכחה: נוכיח חוקיות ולאחר מכן אופטימליות.

<u>חוקיות</u>: הפתרון מכיל את a_1, a_n. בנוסף, לכל $1 < i \leq m$ מתקיים כי $b_i - b_{i-1} \leq N$ מהתנאי השני להוספת a_i. האלגוריתם עוצר לאחר n איטרציות כשסיים לעבור על כל התחנות ולכן עוצר ומחזיר פתרון חוקי.

<u>אופטימליות</u>: מהנתון קיים פתרון אופטימלי $c_{m+1}, \ldots, c_{m'}$ כך ש-$B = \{b_1, \ldots, b_m, c_{m+1}, \ldots, c_{m'}\}$ אבל מהיות B פתרון חוקי ו-$b_m = a_n$ נובע כי $m' = m$ ולכן B אופטימלי. ■ עכשיו נותר לנו רק להוכיח את הלמה.

הוכחה: (הלמה) נוכיח באינדוקציה על k.

<u>בסיס</u>: $k = 1$ במקרה זה בהכרח קיים פתרון אופטימלי שמכיל את $b_1 = a_1$ כי כל פתרון חוקי מכיל את a_1 בהתחלה.

<u>שלב</u>: נניח נכונות עבור k נוכיח נכונות עבור $k+1$.

מהנחת האינדוקציה, קיים פתרון אופטימלי $c_{k+1}, \ldots, c_{m'}$ כך ש-$C = \{b_1, \ldots, b_k, c_{k+1}, \ldots, c_{m'}\}$ אופטימלי. נראה כי הפתרון $C' = \{b_1, \ldots, b_k, b_{k+1}, c_{k+2}, \ldots, c_{m'}\}$ חוקי, שכן אם הוא חוקי, אזי הוא בגודל m' ולכן אופטימלי.

מהיות B פתרון חוקי, מתקיים לכל $1 \leq i \leq k$ כי $b_{i+1} - b_i \leq N$ וכן מהיות C חוקי מתקיים כי לכל $k+1 \leq j \leq m' - 1$ מתקיים כי $c_{j+1} - c_j \leq N$ וכן $c_{k+1} - b_k \leq N$. בנוסף, $c_{m'} = a_n, b_1 = a_1$ מחוקיות C.

על כן מספיק להראות כי $c_{k+2} - b_{k+1} \leq N$. מתקיים כי

$$c_{k+2} - b_{k+1} = c_{k+2} - c_{k+1} + c_{k+1} - b_{k+1} \leq N + c_{k+1} - b_{k+1}$$

נראה כי $b_{k+1} \geq c_{k+1}$. מאופן פעולת האלגוריתם אנו יודעים כי b_{k+1} הינה התחנה הרחוקה ביותר אליה ניתן להגיע מ-b_k, לכן $b_{k+1} \geq c_{k+1}$ מכאן

$$c_{k+2} - b_{k+1} \leq N + c_{k+1} - b_{k+1} \leq N$$

כרצוי. מכאן C' פתרון חוקי ולכן גם אופטימלי. ■ נותר לנו לנתח את זמן ריצת האלגוריתם. אנו עוברים על n איברים ובכל פעם מבצעים מספר חסום ל פעולות אריתמטיות לכן $O(1)$. לכן סך הכל $O(n)$.

הערה. הטענה שהוכחנו עכשיו אמנם הוכחה באופן ספציפי ביחס לבעיה והאלגוריתם, אבל נראה שזו טכניקה שימושית להוכחת נכונות של אלגוריתמים חמדניים. ננסה להכליל זאת.

5.2 סכימה כללית להוכחת נכונות של אלגוריתם חמדן

יהי אלגוריתם חמדן B. נוכיח את נכונותו באופן הבא:

1. נוכיח חוקיות של B.
2. נוכיח טענת רישא אינדוקטיבית על B: לכל $1 \leq k \leq m$ קיים פתרון אופטימלי C שמסכים עם B על k האיברים הראשונים, כאשר $m = |B|$.
3. הוכחת טענת האינדוקציה:
   (א) <u>בסיס</u>: $k = 0$, טריוויאלי. כאשר ניזהר, אם הצעד לא תקף על הבסיס, נוכיח את המקרה $k = 1$.
   (ב) <u>שלב</u>: אם $k > m$ אין מה להוכיח, כי טענת האינדוקציה טוענת על $k \leq m$. נניח כי $k \leq m$ אזי מהנחת האינדוקציה קיים פתרון אופטימלי C שמסכים עם B על $k-1$ האיברים הראשונים. נבנה מ-C פתרון חוקי חדש C' שמסכים עם B על k האיברים הראשונים. יתכן שנצטרך לשנות עוד איברים מלבד האיבר ה-k, אך כל עוד לא שינינו את האיברים עד האיבר ה-$k-1$, קיבלנו פתרון חוקי כרצוי. נותר להוכיח ש-C' אופטימלי וחוקי, זה החלק שהאלגוריתם עצמו והבעיה הופכים לרלוונטים להוכחה, נשתמש באופטימליות של C ובאופן פעולת B.
4. מסקנה: גם עבור $k = m$ קיים פתרון אופטימלי C שמכיל את m האיברים הראשונים של B ולכן B אופטימלי (מעקרון בלמן).

הערה. הסכימה הנ"ל מניחה שקיים פתרון אופטימלי לבעיה. כל עוד מרחב הפתרונות סופי, זה נכון. אך מה קורה כאשר המרחב איננו סופי? במקרים אלה, נוכל להעזר בשיקולים מהחשבון האינפיניטיסימלי ולהראות קיום של מקסימום או מינימום באמצעות רציפות של פונקציית ערך ומשפט וירשטראס, כמו שעשינו בבעיית התרמיל השברי.

5.3 מציאת עץ פורש מינימלי

קלט גרף קשיר ולא מכוון $G = \langle V, E \rangle$ בעל צלע אחד לפחות ו-$w : E \to \mathbb{R}$.

פלט MST.

אלגוריתם 6 אלגוריתם חמדן לפתרון בעיית MST (Kruskal)

- עיבוד מקדים: נסמן $|V| = n$ ו-$|E| = m$ נמיין את הצלעות בסדר עולה לפי משקל $e_1, \ldots, e_m$ כלומר $w(e_i) \le w(e_{i+1})$ לכל $1 \le i \le m-1$.
- אתחול: נאתחל $T = \emptyset$ קבוצת הצלעות שנחזיר.
- איטרציה: נעבור על הצלעות לפי הסדר, ולכל i נוסיף את e_i ל-T אם"ם $T \cup \{e_i\}$ חסר מעגלים.

נוכיח את נכונות האלגוריתם.

טענה. (חוקיות) האלגוריתם מחזיר פתרון חוקי לבעיה.

הוכחה: יוכח בתרגיל הבית. ■ נותר לנו להראות אופטימליות. כיצד? עם הסכימה כמובן. נסמן את צלעות הפלט ב-$t_1, \ldots, t_{n-1}$ כאשר הן ממוינות לפי משקל כלומר $w(t_i) \le w(t_{i+1})$ לכל $1 \le i \le n-2$. נסמן ב-T_k את תת הגרף שמכיל רק את הצלעות $\{t_1, \ldots, t_k\}$.

למה. (טענת האינדוקציה על Kruskal) לכל $1 \le k \le n-1$ קיים עץ פורש מינימלי שצלעותיו מכילות את T_k.

מסקנה. T עץ פורש מינימלי.

הוכחה: מהטענה קיים עת פורש מינימלי המכיל את צלעות T, אבל בעץ פורש יש בדיוק $n-1$ צלעות ולכן צלעות העץ הפורש הנ"ל הן בדיוק הצלעות ב-T ולכן T עפ"ם. ■

הוכחה: (הלמה) באינדוקציה על k.

בסיס: $k = 0$ מקרה טריוויאלי כי $T_0 = \emptyset$ וכל קבוצה מכילה אותה, בפרט עץ פורש מינימלי.

שלב: נניח כי קיים עץ פורש מינימלי S כך ש-$T_{k-1} \subseteq S$ ונראה כי קיים עץ פורש מינימלי S' כך ש-$T_k \subseteq S'$.

אם $t_k \in S$ נוכל להגדיר $S' = S$ וסיימנו. אחרת $t_k \notin S$. נביט ב-$S \cup \{t_k\}$ הגרף הזה התקבל על ידי הוספת צלע לעץ S ולכן בהכרח סגרנו בו מעגל פשוט יחיד המכיל את t_k. תהי e צלע הנמצאת במעגל אך לא ב-T. נחלק למקרים.

- $w(e) < w(t_k)$. נביט בשלב בו מוסיפים את t_k ל-T. מכך ש-$w(e) < w(t_k)$ האלגוריתם ניסה להוסיף את הצלע e עוד קודם אבל לא עשה זאת כי סגרה מעגל עם הצלעות $t_1, \ldots, t_j$ עבור $1 \le j \le k-1$ כלשהו. לכן e סוגרת מעגל גם עם הצלעות הנ"ל ב-S בסתירה לכך ש-S חסר מעגלים. לכן מקרה זה לא יכול להתקיים.
- $w(e) > w(t_k)$ במקרה זה נוכל נסיר את e ונקבל $S' = S \cup \{t_k\} \setminus \{e\}$ זהו עץ פורש ויתרה מזאת מתקיים כי

  $$w(S \cup \{t_k\} \setminus \{e\}) = w(S) + w(t_k) - w(e) < w(S)$$

  בסתירה למינימליות S. לכן מקרה זה גם לא יכול להתקיים.
- $w(e) = w(t_k)$. לכן העץ $S' = (S \cup \{t_k\}) \setminus \{e\}$ הוא עץ פורש המכיל את t_k ומתקיים $w(S') = w(S)$ ולכן הוא גם מינימלי.

ובזאת סיימנו את הוכחת הלמה. ■

6 תרגול 3 - מטרואידים

6.1 מבוא

נציג למעשה הגדרה של אובייקט קומבינטורי כלשהו ונבין איך הוא מתקשר אלינו.

הגדרה. מטרואיד הוא זוג $M=\langle S,I\rangle$ כאשר S קבוצה סופית ו-$I\subseteq 2^S$, המקיים את התכונות הבאות:

(i) : $I\neq\emptyset$.

(ii) : תורשתיות: אם $A\in I$ וב-$B\subseteq A$ אז גם $B\in I$.

(iii) : החלפה: אם $A,B\in I$ ו-$|A|>|B|$ אזי קיים $a\in A\backslash B$ כך ש-$B\cup\{a\}\in I$.

מסקנה. מתכונה (ii) נקבל כי $\emptyset\in I$ ולכן בתנאי (i) יכולנו לדרוש כי $\emptyset\in I$.

הערה. נוכל לחשוב על S ככל האיברים שנוכל להכניס ו-I אוסף כל הפתרונות החוקיים.

דוגמה. באלגוריתם קורסקל, $S=E$ ו-I כל הפתרונות החוקיים שהם קבוצה חוקית של צלעות.

דוגמאות

1. המטרואיד השלם: נגדיר $M_1=\langle S_1,I_1\rangle$ המוגדר על ידי $S_1=\{1,2\},I_1=\{\{1\},\{2\},\{1,2\}\}\cup\{\emptyset\}=2^S$ זהו מטרואיד. אך ללא הקבוצה הריקה זה לא מטרואיד באופן כללי, תמיד הזוג $\left\langle S,2^S\right\rangle$ הוא מטרואיד ונקרא "המטרואיד השלם".
2. $M_2=\langle S_2,I_2\rangle$ עם $S_2=[4]$ ו-$I_2=\{\emptyset,\{1\},\{2\},\{3\},\{4\},\{1,2\},\{3,4\}\}$ לא מטרואיד כי לא מתקיימת תכונת ההחלפה, שכן אם ניקח את $\{2\},\{3,4\}$ הקבוצות $\{2,3\},\{2,4\}\notin I_2$.

הגדרה. המטרואיד הווקטורי. יהי V מרחב הווקטורי כלשהו נסמן ב-$M_V=\langle S_V,I_V\rangle$ כאשר $S_V\subseteq V$ ו-I_V היא כל תתי הקבוצות של S_V שהן בת"ל.

טענה. יהי V מרחב ווקטורי אזי $M_V=\langle S_V,I_V\rangle$ הוא מטרואיד.

הוכחה: אנו נוכיח בכיתה. לעת עתה, אנו נשתמש בכך שלכל קבוצות בת"ל בגודל שונה ניתן להעביר ווקטור מהקבוצה הגדולה לקטנה כך שהיא נשארת בת"ל ולכן תשמר תכונת ההחלפה. ■

6.2 האלגוריתם החמדן הגנרי למטרואידים

בעיה. יהי מטרואיד $M=\langle S,V\rangle$ ותהי פונקציית משקל $w:S\to\mathbb{R}_+$ כך ש-$\forall T\subseteq S:w(T)=\sum\limits_{e\in T}w(e)$. נרצה למצוא $T\in I$ עבורה $w(T)$ מקסימלי או מינימלי.

נציג את האלגוריתם:

אלגוריתם 7 האלגוריתם החמדן הגנרי למטרואידים

1. נאתחל $T=\emptyset$.
2. נמיין את איברי S מהגדול לקטן. $\Theta(n\log n)$
3. נעבור על כל $s\in S$ מהגדול לקטן.
   (א) אם $T\cup\{s\}\in I$ נוסיף את s ל-T. נסמן סיבוכיות זו ב-$f(n)$.

הערה. ניתן לקבל גם גודל מינימלי אם נשנה את סדר מיון האיברים.

דוגמה. זו הכללה של האלגוריתם קורסקל והאלגוריתם שראינו בכיתה על ווקטורים בת"ל עם משקל מקסימלי.

טענה. אלגוריתם זה אופטימלי.

הוכחה: נראה בכיתה. ■

טענה. שני פתרונות אופטימלים הם באותו גודל.

הוכחה: אחרת מתכונת ההחלפה היינו להעביר איבר מהגדולה לקטנה ולקבל פתרון טוב יותר, בסתירה לאופטימליות של שני הפתרונות. ■

6.2.0.1 זמן הריצה של האלגוריתם נקבל כי זמן הריצה הוא $\Theta\left(n\left(\log n+f\left(n\right)\right)\right)$ ואם $f\left(n\right)$ קטן יחסית מ-$\log n$ נקבל כי הוא אכן יעיל.

6.3 המטרואיד הגרפי

הגדרה. יהי גרף קשיר לא מכוון $G=\left(V,E\right)$. נגדיר את המטרואיד הגרפי להיות $M_G=\langle S_G,I_G\rangle$ כאשר $S_G=E$ ו-$I_G=\{A\subseteq S_G\mid$ הוא יער $G_A=\langle V,A\rangle\}$.

טענה. המטרואיד הגרפי הוא מטרואיד.

הוכחה: נוכיח כל אחד מהתנאים.

(i) : $\emptyset\in I_G$ כי $\langle V,\emptyset\rangle$ היא יער.

(ii) : תורשתיות: יהי $F\in I_G$ ותהי $F'\subseteq F$ אזי משום שאין מעגל ב-F, אין מעגל גם ב-F'.

(iii) : החלפה: יהיו $A,B\in I_G$ כלומר $\langle V,A\rangle,\left(V,B\right)$ יערות, כך ש-$|B|>|A|$. נרצה להוכיח שקיימת צלע $e\in A\backslash B$ כך ש-$\langle V,B\cup\{e\}\rangle$ הוא יער. דבר כזה לא מאוד טריוויאלי, אך בואו ננסה לחשוב על זה. מה יקרה עם נוסיף צלע? נסגור מעגל אולי, מתי בדיוק? כאשר הוספנו צלע לשני קודקודים באותו רכיב קשירות. על כן, נרצה להוסיף צלע בין שני רכיבי קשירות שונים ב-G_B. מכאן עלינו להוכיח כי אכן קיימת צלע ב-A שמחברת שני רכיבי קשירות שונים ב-G_B.

תובנה: ב-G_A יש פחות רכיבי קשירות מ-G_B שכן יש יותר צלעות ב-A מב-B וכל הוספת צלע רק מקטינה או את מספר רכיבי הקשירות או סוגרת מעגל, אך כידוע, אין מעגלים בשני הגרפים כי מדובר ביערות, ולכן כל הוספת צלע שם רק מקטינה באחד את מספר רכיבי הקשירות.

נרצה למצוא צלע ב-A שמחברת בין שני רכיבי קשירות ב-G_B.

נניח בשלילה שאין אף רכיב קשירות ב-G_A שחותך יותר מרכיב קשירות אחד ב-G_B כאשר הכוונה בחותך היא שקיימים שני קודקודים ברכיב קשירות זה ששייכים לשני רכיבי קשירות שונים. כלומר כל רכיב קשירות ב-G_A מוכל ברכיב קשירות יחיד ב-G_B, שכן כל קודקודיו גם הם חלק מהגרף G_B והם חייבים להיות שייכים לרכיב קשירות אחד ב-G_B, אחרת היינו מקבלים שהרכיב ב-G_A כן חותך שני רכיבי קשירות שונים ב-G_B.

מכאן נרצה לטעון שמספר רכיבי הקשירות ב-G_B קטן יותר ממספר רכיבי הקשירות ב-G_A שכן כל רכיב קשירות ב-G_A מוכל ברכיב קשירות יחיד ב-G_B ולכן בכל רכיב קשירות ב-G_B יש לכל הפחות תת גרף שהא רכיב קשירות אחד ב-G_A ועל כן מספר רכיבי הקשירות ב-G_A גדול או שווה ממספר רכיבי הקשירות ב-G_B, שזו סתירה לתובנה שלנו. נראה זאת באופן פורמלי.

נגדיר פונקציה מרכיבי הקשירות ב-G_A לרכיבי הקשירות ב-G_B שתאמר לנו באיזה רכיב הקשירות ב-G_B נמצא הרכיב הנוכחי ב-G_A. אנו מקבלים שהפונקציה היא על ומכאן נסיק כי מספר רכיבי הקשירות ב-G_A גדול או שווה מרכיבי הקשירות ב-G_B, שזו סתירה.

מכאן אנו מסיקים כי יש רכיב קשירות ב-G_A שחותך שני רכיבי קשירות שונים ב-G_B, האם סיימנו? לא, שכן לא בהכרח קיימת ביניהם צלע אלא רק מסלול! נראה שבהכרח קיימת כזו. אם אין אף צלע ב-A שחוצה שני רכיבי קשירות ב-B, אז כל רכיב קשירות ב-A מוכל ברכיב קשירות ב-G_B, שכן אחרת, אם הוא היה מוכל בשני רכיבי קשירות שונים, הייתה צלע שמחברת בין שני קודקודים משני רכיבים שונים בתוך רכיב הקשירות G_A. וכאן אנו מקבלים שוב סתירה למה שראינו קודם. על כן קיימת צלע כנ"ל מ-A, נוסיפה ל-B ונקבל ש-G_B יער. ■

מסקנה. קרוסקל נותן פתרון אופטימלי.

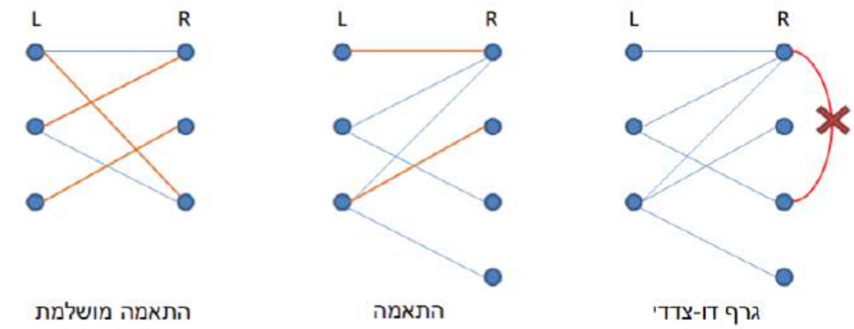
הוכחה: נניח כי הוכחנו את נכונותו האלגוריתם הגנרי למטרואיד. נריץ אותו על המטרואיד הגרפי המתאים לגרף הנתון, ונקבל קבוצה T עם משקל מינימלי שהיא בגודל מקסימלי, יער בגודל מקסימלי בגרף קשיר הוא עץ פורש ולכן קיבלנו עץ פורש מינימלי. ■

6.4 גרפים דו צדדיים וזיווגים מושלמים

נחזור לימי העבר בהם למדנו דיסקרטית.

הגדרה. גרף דו צדדי (דו"צ) הוא שלשה (L,R,E) כך ש-$L\cap R=\emptyset$ וגם $E\subseteq\{\{l,r\}\mid l\in L, r\in R\}$.

דוגמה. נביט באיור הבא להמחשה:



איור 1: המחשה לגרף דו צדדי

הגדרה. יהי גרף דו צדדי (L,R,E) זיווג הוא $E'\subseteq E$ כך שאין אף קודקוד ב-$L\cup R$ שבו נוגעת יותר מצלע אחת ב-E'.

הערה. בזיווג לכל קודקוד יש דרגה שהיא לא היותר 1.

הגדרה. התאמה מושלמת או זיווג מושלם ב-$\langle L,R,E\rangle$ היא התאמה (פונקציה) הנוגעת בכל הקודקודים ב-$L\cup R$.

6.4.1 מטרואיד השידוכים

רה. נגדיר את מטרואיד השידוכים על הגרף $G=\langle L,R,E\rangle$ על ידי $M=\langle S,T\rangle$ עבור $S=L$ ו-$I=\left\{L'\subseteq L \mid \begin{array}{c}\exists R'\subseteq R, R'\subseteq E:\\ \text{הוא זיווג מושלם } \langle L',R',E'\rangle\end{array}\right\}$.

טענה. מטרואיד השידוכים מוגדר היטב, דהיינו הוא מטרואיד.

הוכחה: נוכיח כל אחת מהתכונות.

תורשתיות: כי לכל $A\in I$ יש השלמה להתאמה מושלמת ולכן לכל תת קוצה שלה $B\subseteq A$ יש השלמה להתאמה מושלמת.

התכונה הבאה קשה יותר להוכחה.

החלפה: יהיו $A,B\in I$ כך ש-$|B|>|A|$ אזי קיימים R_A,R_B,E_A,E_B כך ש-$(A,R_A,E_A),(B,R_B,E_B)$ זיווגים מושלמים. מטרתנו היא במידה מסויימת להרוס את הזיווג הקיים בקבוצה הקטנה ולסדר את הצלעות מחדש כך שלאחר ההוספה עדיין יהיה זיווג מושלם. נסתכל על האיחוד של שני הגרפים, זאת מכיוון שפעמים רבות זה עוזר, כלומר נביט בגרף $G'=G_A\cup G_B$. נסמן ב-M_B את הזיווג המושלם המתאים ל-B וב-M_A את הזיווג המושלם המתאים ל-A. כלומר $G'=(L,R,M_A\cup M_B)$.

מתקיים

1. ב-G' כל הקודקודים מדרגה קטנה שווה 2, שכן הם לכל היותר ב-M_A,M_B.
2. מ-1 נובע כי כל רכיב קשירות C יכול להיות אחד מהבאים:
   (א) C הוא מעגל פשוט - כלומר דרגת כל קודקוד היא בדיוק 2.
   (ב) C הוא מסלול פשוט - כלומר דרגת כל קודקוד פנימי היא 2 והסוף וההתחלה הם מדרגה 1.
3. בכל רכיב קשירות ב-G' הצלעות צריכות להיות לסירוגין מ-M_A ל-M_B או להפך, שכן אחרת נקבל שיש קודקוד מדרגה 2 שהוא בזיווג מושלם, וזו סתירה להגדרת הזיווג המושלם.

4. נסמן ב-$E_C\left(M_A\right)$ את מספר הצלעות של הזיווג M_A ברכיב הקשירות C וב-$E_C\left(M_B\right)$ את מספר הצלעות של M_B ברכיב הקשירות C. אז מתכונה 3 נובע כי לכל רכיב קשירות C מתקיים כי $\left|E_C\left(M_A\right)-E_C\left(M_B\right)\right| \leq 1$.

מהיות $|B|<|A|$ נובע כי $|M_B|<|M_A|$, ומשיקולי ספירה, חייב להיות רכיב קשירות C שעבורו $E_C\left(M_A\right)>E_C\left(M_B\right)$.

נבחין כי C חייב להיות מסלול פשוט באורך אי זוגי אשר מתחיל בצלע ב-M_A ומסתיים בצלע השייכת ל-M_A, שכן אחרת, אם הוא מעגל או מסלול באורך זוגי, נובע כי $E_C\left(M_A\right)=E_C\left(M_B\right)$.

מסלול זה, מכיוון שהינו מסלול באורך אי זוגי, בהכרח מתחיל או נגמר בקודקוד מ-L. נסמן קודקוד זה ב-x, אזי $x\in A\backslash B$, שכן אחרת, הקודקוד הבא לאחר x היה מדרגה 2 והוא קודקוד מ-B, שכן הצלעות הן לסירוגין מ-A, B. סתירה.

נביט בשידוך $B'=B\cup\{x\}$ המוגדר באופן הבא: את כל הצלעות שהיו ב-M_B נשאיר, מלבד הצלעות שהיו ב-C. במקום צלעות אלה, נכלול את כל הצלעות שהיו שייכות ל-A ברכיב זה. כלומר $M_{B'}=\left(M_B\backslash E_C\left(M_B\right)\right)\cup E_C\left(M_A\right)$. עבור כל רכיב קשירות אנחנו מקבלים שתכונת הזיווג נשארה נכונה. עבור רכיב הקשירות C זה גם מתקיים שכן עבור הקודקוד הראשון יש את הצלע מ-M_A. לאחר מכן הקודוקד הבא הוא קודקוד מ-B שממנו יוצאת צלע מ-M_A. אל הקודוקד הבא נגיע לאחר מעבר בצלע של M_B ולכן מה שיתאים לקודקוד הבא מ-B הוא בוודאות קודקוד חדש שלא מיפינו קודם.

לכן $M_{B'}$ זיווג מושלם. ∎

חלק III

תכנון דינאמי - Divide And Conquer

בשונה מהעקרון החמדני, בו אנו בוחרים כמה שיותר, כאן העקרון חדש לגמרי.

נחלק את הבעיה הנתונה לתת בעיות, נפתור את תת הבעיות ונמזג את פתרונן לפתרון הבעיה כולה. זאת אסטרטגיית "הפרד ומשול".

7 מבוא - חישוב פונקציה המוגדרת באופן רקורסיבי

בעיה. נתונה פונקציה $T\left(a,b\right)$ של שני פרמטרים טבעיים המוגדרת באופן הבא:

$$T\left(a,b\right)=\begin{cases} 1 & a=0\vee b=0\\ T\left(a-1,b\right)+T\left(a,b-1\right) & \text{אחרת} \end{cases}$$

מהו $T\left(2021,2021\right)$?

פתרון. (נאיבי) נריץ את הנוסחא רקורסיבית עד שנסיים:

$$\begin{array}{ccccc} & & (2021,2021) & & \\ & (2020,2021) & & (2021,2020) & \\ & \cdots & \cdots & \cdots & \\ \vdots & & & & \vdots \\ (\cdot,0) & & \cdots & & (0,\cdot) \end{array}$$

העץ הזה הוא עץ בינארי מלא בעומק 2021 ולכן זמן הריצה הוא $\Omega\left(2^{2021}\right)$. יקר מדי. נבחין כי בפועל יש חזרה על בעיות ולכן נוכל לחסוך זמן רב.

שאלה מה הן תת הבעיות השונות שנצטרך לפתור לאורך הריצה של הרקורסיה?

נבחין כי כל תת הבעיות האלה הן חישוב של $T\left(a,b\right)$ עבור $0\leq a,b\leq 2021$ ולכן יש $2022^2<\left(2048\right)^2=2^{22}<<2^{2021}$ אפשרויות.

שאלה איך נארגן את כל תתי הבעיות כך שנפתור כל תת בעיה פעם אחת.

השערה. כדי לפתור בעיה חלקית רק פעם אחת, נמלא טבלא[1].

אם כך, נמלא טבלא: השורה האחרונה והעמודה האחרונה מתאימות ל-$a=0 \vee b=0$ ולכן יתמלאו ב-1. כל שאר האיברים הם סכום של מה שמתחתיהם פלוס מה שמשמאלם:

2021	1								
	1				$\leftarrow$	$\cdot$			
	$\vdots$					$\downarrow$			
T	1	3	6						
	1	2	3	4					
0	1	1	1	1	$\ldots$	1	1	1	1
	0								

על כן אנו מקבלים בדיוק משולש פסקל הפוך, ולכן נוכל להסיק כי $T\left(a,b\right)=\binom{a+b}{a}$, אינטואיטיבית, עלינו לחשוב על מספר המסלולים מהנקודה $\left(a,b\right)$ לשורה אחרונה או לעמודה אחרונה, על כן, עלינו לבחור מתוך $a+b$ צעדים אפשריים, בדיוק a תזוזות שמאלה או לבחור b ירידות שזה שקול כמובן. מכיוון שזמן מילוי הטבלא, על ידי מילוי איברי האלכסון ויצירת משולש פסקל הוא $\Theta\left(n^2\right)$ נקבל כי סך הכל זמן ריצת האלגוריתם הוא $\Theta\left(n^2\right)$.

8 בעיית ניתוב משימות

8.0.0.1 סיפור מסגרת פריט עובר תהליך ייצור. יש שני פסי ייצור זהים. על כן פס מספר תחנות עבודה. המחירים של לעבור מתחנה לתחנה שונים זה מזה. רוצים למצוא מסלול לפריט בין תחנות העבודה כדי למזער את המחיר הכולל של תהליך הייצור.

קלט גרף

$$\begin{array}{cccccc}
 & u_1\rightarrow & u_2\rightarrow & \ldots & u_{n-1} \\
\nearrow & x_1 & x_2 & x_3 & \downarrow \\
S & & & & f \\
\searrow & y_2 & y_3 & \ldots y_{n-1} & \uparrow \\
y_1 & d_1\rightarrow & d_2\rightarrow & \ldots & d_{n-1}
\end{array}$$

בו אנו מתחילים מ-S ועוברים בין שני הפסים הייצור U ו-D. על כן הקלט הוא

$$x_1,x_2,\ldots,x_n$$

$$y_1,y_2,\ldots,y_n$$

$$a_2,\ldots,a_{n-1}$$

$$b_2,\ldots,b_{n-1}$$

כאשר הפריט יכול לעבור בין התחנות הסמוכות ומחיר המעבר הוא לפי a,b.

פלט מסלול מ-S ל-f בעל מחיר מינימלי.

השערה. אלגוריתם מזן טבעי: בכל פיצול נעדיף את הכיוון הזול ביותר.

[1] המילה האנגלית העתיקה לטבלא היא program.

בעיה. לא עובד. נבחר
$$\begin{array}{cc} 1 & 100 \\ \nearrow & \searrow \\ \searrow & \nearrow \\ 10 & 1 \end{array}.$$

האם יש דרך אחרת? אנו מחפשים מסלול קצר ביותר בין שני קודקודים בגרף, הביע ניתנת לפתרון על ידי האלגוריתם של $Dijkstra$. במקרה זה האלגוריתם הנ"ל רץ בזמן $\mathcal{O}\left(n\log n\right)=\mathcal{O}\left(\left|V\right|\log\left|V\right|\right)$. אנו ניתן אלגוריתם דינאמי הרץ בזמן $\mathcal{O}\left(n\right)$.

בכל זאת, ניתן אלגוריתם נאיבי.

השערה. (נאיבי) לעבור כל כל המסלולים. גודל מרחב הפתרונות החוקיים של הבעיה הוא 2^{n-1}. כלומר זמן הריצה יהיה $\Omega\left(2^{n}\right)$. לכן המעבר על כל האפשרויות יקר מדי.

כיצד נפתור זאת? נבחין כי ניתן לחלק את המסלול לשני חצאים $u_{\frac{n}{2}},d_{\frac{n}{2}}$, ולפתור את הבעיה על ידי יציאה מכל אחת מהיציאות $d_{1},u_{1},u_{n/2},d_{n/2}$. בנוסף, תת מסלול של מסלול אופטימלי הוא גם אופטימלי (אחרת היינו יכולים לקבל מסלול כולל טוב יותר). לעקרון זה יש שם:

8.0.0.2 עקרון $Bellman$ תת פתרון של פתרון אופטימלי הוא עצמו אופטימלי.

בהנתן הבחנה זו, ניתן לפצל בעיה בגודל n ל-4 בעיות בגודל $\frac{n}{2}$ ולפתור על ידי מעבר על כל האפשרויות ב-$\Theta\left(2^{\frac{n}{2}}\right)$ ולא ב-$\Theta\left(2^{n}\right)$. נוכל להמשיך לפצל ונגיע ל-16 בעיות באורך $\frac{n}{4}$. נוכל להוריד את זמן הריצה אם נחליט לבנות נוסחת נסיגה $T\left(n\right)=4T\left(\frac{n}{2}\right)+\Theta\left(1\right)$ שפתרונה $\Theta\left(n^{2}\right)$. יחד עם זאת, החלוקה ב-2 בעייתית כי לא תמיד המספר זוגי.

נרצה לפצל את הבעיה בצורה מקומית. נחלק את הבעיה הגדולה לשתי תתי בעיות לפי הצעד הראשון במסלול. אם בחרנו ללכת מ-s ל-u_{1} נגיע לתת הבעיה הבאה:

$$\begin{array}{cccccccc} & & \nearrow & u_{2}\rightarrow & u_{3} & \ldots & u_{n-1} & \searrow \\ S & \rightarrow & u_{1} & & & & & f \\ & & \searrow & d_{2} & \rightarrow d_{3} & \ldots & d_{n-1} & \nearrow \end{array}$$

דיאגרמה רקוסיבית זו איננה משפרת את זמן הריצה, להפך, היא הופכת אותו לאקספוננציאלי, אבל היא כן מוסיפה לאינטואיציה שלנו.

נרצה להבין מהי התלות של המחיר האופטימלי של הבעיה הגדולה במחירים של תת הבעיות האלה. נסמן ב-$p^{\star}$ את מחיר הבעיה הגדולה, ב-$p_{u}\left[1\right]$ את המחיר האופטימלי להגיע מ-u_{1} ל-f וב-$p_{d}\left[1\right]$ להגיע מ-d_{1} ל-f. אז $p^{\star}=\min\left\{ x_{1}+p_{u}\left[1\right],y_{1}+p_{d}\left[1\right]\right\}$.

נמשיך לפצל, ונרצה להבין לאיזה תת בעיות נגיע במהלך הפיצולים הבאים. התשובה היא שעלינו להגיע מכל אחת מהתחנות $d_{1},\ldots,d_{n-1},u_{1},\ldots,u_{n-1}$ ל-f.

נסמן ב-$p_{u}\left[k\right]$ עבור $k=1,\ldots,n-1$ את המחיר האופטימלי להגיע מ-u_{k} ל-f. וב-$p_{d}\left[k\right]$. את המחיר האופטימלי להגיע מ-d_{k} ל-f.

נכתוב את נוסחת הרקורסיה המבטאת את התלות של מחיר של תת בעיה נתונה במחיר שתי תתי הבעיות שהיא מתפצלת אליהן:

$$p_{u}\left[k\right]=\begin{cases} x_{n} & k=n-1\\ \min\left\{ x_{k+1}+p_{u}\left[k+1\right],a_{k}+p_{d}\left[k+1\right]\right\} & k<n-1 \end{cases}$$

שכן מעבר בין פסי ייצור הוא על פי a,b.

8.1 האלגוריתם

נבנה טבלא שנמלא מסדר הפוך.

$$\begin{array}{ccccccc} & 1 & k & k+1 & \dots & n-1 & \\ u & p^\star & p_u[k] & \cdot & \dots & x_n & f \\ d & p^\star & p_d[k] & \cdot & \dots & y_n & f \end{array}$$

מעבר מ-· ל-$p_u[k]$ יחושב ב-$\mathcal{O}(1)$ כי העמודה ה-$k+1$ כבר חושבה.

אנו ממלאים את הטבלא עמודה עמודה מימין לשמאל. כאשר בעמודה ה-$n-1$ נרשום את הווקטור $\begin{pmatrix} x_n \\ y_n \end{pmatrix}$. בהנתן העמודה ה-$k+1$ נוכל למלא את העמודה ה-k לפי נוסחת הרקורדיה הכללית בזמן $\mathcal{O}(1)$. סך הכל, נמלא את כל תאי הטבלא בזמן $\mathcal{O}(1)$, יש בטבלא $\mathcal{O}(n)$ תאים, לכן נוכל למלא את הטבלא כולה בזמן $\mathcal{O}(n)$.

אם כך, נמלא את הטבלא כולה ונחזיר את $p^\star$. כאן החזרנו רק את המחיר האופטימלי! מה לגבי המסלול עצמו?

כדי למצוא את המסלול האופטימלי, נזכור בזמן מילוי כל תא בטבלא, מהו הכיוון הנכון לבחירה בשלב הזה, כאן אנו הולכים משמאל לימין.

9 תרגול 4 - תכנון דינאמי

פרק זה משלב בין רקורסיה לאקסטרה זכרון בצד.

דוגמה. נגדיר $\text{fibo}(n+2) = \text{fibo}(n+1) + \text{fibo}(n)$ עם $\text{fibo}(1) = 1 = \text{fibo}(2)$. נרשום אלגוריתם לחישוב האיבר ה-n:

Algorithm 8 Fibo (n)

1: if $n \in \{0,1\}$: return 1

2: return $\text{fibo}(n-1) + \text{fibo}(n-2)$

האלגוריתם עובד, אבל מה הבעיה בו? אם נשרטט עץ רקורסיה נבחין כי יש הרבה חזרות של איברים בסדרה ולכן נוכל להציע טבלא, האם יש כאן זכרון נוסף? זה נראה שלא, אבל בפועל אנו ממשים זאת עם הרבה זכרון נוסף. מעבר לכך זמן הריצה שלו הוא $\Theta(\varphi^n)$, אקספוננציאלי. על כן, נציע אלגוריתם טוב יותר:

Algorithm 9 Fibo (n)

1: if $n < 3$: return 1

2: $cur, prev = 1, 1$

3: for i in $range(3, n)$:

4: cur,prev = cur+prev, cur

5: return cur

יש לנו כאן מעט זכרון עזר. אלגוריתם זה לעומת האלגוריתם הקודם, רץ ב-$\mathcal{O}(n)$. כאן לא השתמשנו בטבלא.

9.1 לוח משימות תכנות

9.1.0.1 סיפור מסגרת עליכם לבחור את משימות התכנות שלכם ל-n השבועות הקרובים. לכל אחד מהשבועות יש שתי משימות אפשריות: אחת בלחץ עבודה גבוה ואחת בלחץ עבודה נמוך. עבור משימה בלחץ עבודה גבוה בשבוע ה-i נתוגמל ב-$h_i > 0$ שקלים, עבור משימה בלחץ עבודה נמוך בשבוע ה-i נתוגמל ב-$l_i > 0$ שקלים. אם בוחרים לנוח בשבוע כלשהו לא מקבלים תשלום עבור שבוע זה, כמו כן, אם נבחרה משימה בלחץ גבוה לשבוע מסויים יש להערך לכך על ידי מנוחה בשבוע הקודם (התנאי לא חל על השבוע הראשון). בנו לוח משימות רווחי ביותר ל-n שבועות בהנתן מחירים המשימות. עלינו למקסם את סכום הכסף שאפשר לקבל מביצוע המשימות תכנותיות ב-n השבועות.

קלט $\{(l_i h_i)\}_{i\in[n]}$ כך ש-l_i היא כמות הכסף שנקבל עבור העבודה הקלה בשבוע ה-i ו-h_i עבור המשימה הקשה.

פלט הכמות המקסימלית שאפשר לקבל עד השבוע ה-n.

איך נפתור בעיה זו? נפריד ונמשול. אם אני בשבוע האחרון, לא משתלם לי לנוח, אז משתלם לי או לעשות את המשימה הקלה או לעשות משימה קשה ולנוח שבוע קודם ואז להסתכל על הרווח מלפני שבועיים. נרצה למקסם בכל צעד את הרווח. אם כך, על פי האינטואיציה שצברנו נשאל ונציע נוסחא,

1. <u>אוסף תתי הבעיות</u>: לכל $1 \le i \le n$ מה כמות הכסף המקסימלית עד השבוע ה-i?

2. <u>נוסחת הרקורסיה</u>: נגדיר

$$F(i) = \begin{cases} 0 & i = 0 \\ \max\{h_i, l_i\} & i = 1 \\ \max\{l_i + F(i-1), h_i + F(i-2)\} & i \ge 2 \end{cases}$$

איך נחשב נוסחא כזו? זה בדיוק כמו פיבונאצ'י! נגדיר טבלא!

3. <u>נגדיר טבלא M</u>: בגודל $n+1$ ונמלא אותה לפי נוסחת הרקורסיה בסדר עולה כלומר מ-$i=0$ עד $i=n$. בסוף נחזיר את $M[n]$.

זמן הריצה הוא $\mathcal{O}(n)$.

דוגמה. $\{(10,15),(20,30),(10,50),(60,60)\}$ אזי הטבלא תראה כך

0	15	35	65	125

. אם נרצה לזהות את הבחירות שלנו מה נעשה? בכל שלב שבו אנו מתקדמים נשמור את הבחירה שלנו. כדי לדעת מה בחרנו נתחיל מהתא האחרון, אם בחרנו l, נוסיף l לסוף, נלך תא אחורה, אם בחרנו h נוסיף אותו, נוסיף שנחנו ונחזור שני תאים אחורה וכן הלאה... במקרה שלנו נקבל (h,r,h,l), כאן r מסמל מנוחה.

נרשום אם כך אלגוריתם לפתרון הבעיה:

אלגוריתם 10 חישוב מסלול הבחירות π

- נאתחל רשימה $M = []$ אורך n. נתחיל מהתא ה-$i = n$.

- כל עוד לא הגענו ל-$i \in \{0,1\}$ נוסיף את הבחירה הנתונה לתא ה-i.
  - אם מדובר ב-h
    - נבצע $i \leftarrow i-2$
    - נוסיף r למקום ה-$i-1$.
  - אחרת, אם מדובר ב-l
    - נבצע $i \leftarrow i-1$.

הוכחת נכונות

טענה. האלגוריתם שהצענו אופטימלי.

הוכחה: נעבור על סדר הוספת האיברים לטבלא, נוכיח באינדוקציה על $0 \leq i \leq n$ שבתא ה-i כתוב הערך הנכון המקסימלי.

בסיס: עבור $i = 0$ אסור לעבוד ולכן $M[0] = 0$ ולכן זה המקסימום. עבור $i = 1$ אין לנו הגבלה ולכן לקחנו את המקסימום.

שלב: נניח שהטבלא מלאה נכון בתאים $M[j]$ עבור $0 \leq j < i$.

נתבונן בפתרון האופטימלי עד השבוע ה-i. או שעשינו את העבודה הקלה בשבוע האחרון ואז המקסימום התקבל מ-l_i ועוד המקסימום שאפשר לקבל עד השבוע ה-$i-1$, מהנחת האינדוקציה זה $M[i-1]$, או כנ"ל עם $h_i + M[i-2]$. אלו שתי האפשרויות היחידות העומדות לרשותנו בשבוע ה-i ולכן המקסימום בין שתיהן ייתן את הרווח המקסימלי האפשרי עד שבוע זה, וזה בדיוק $M[i]$. ■

9.2 תת מחרוזת

9.2.0.1 סיפור מסגרת נתונות לנו שתי מחרוזות (מערך של תווים) ועלינו למצוא את המחרוזת המשותפת להן באורך מקסימלי, דהיינו מחרוזת שמופיעה באותו הסדר בשתי המחרוזות, לאו דווקא רציפה, באורך מקסימלי. תמ"א (תת מחרוזת ארוכה)

קלט שתי מחרוזות $X = x_1, \ldots, x_n$, $Y = y_1, \ldots, y_m$.

פלט תת מחרוזת של X ו-Y מקסימלית המתקבלת על ידי מחיקת אותיות בעלת אורך מקסימלי.

דוגמה. $X = ABCD$ ו-$Y = BDC$ אזי BD, BC הן תמ"א.

נבחין כי אם נבחר בכל פעם את האות האחרונה, אנחנו לא מגבילים את עצמנו, ונוכל לבחור את המקסימום מבין שתי האפשרויות. זה מוביל נוסחת הרקורסיה הבאה:

$$F(i,j) = \begin{cases} 0 & i = 0 \lor j = 0 \\ 1 + F(i-1, j-1) & x_i = y_j \\ \max\{F(i-1,j), F(i,j-1)\} & x_i \neq y_j \end{cases}$$

נגדיר את $X^i = x_1, \ldots, x_i$ ו-$Y^j = y_1, \ldots, y_j$ רישות באורך j, i בהתאמה. נגדיר אם כך:

תתי בעיות: לכל $1 \leq i \leq n$ ולכל $1 \leq j \leq m$ נמצא את אורך התמ"א של X^i, Y^j.

איך אנו מחשבים זאת? עץ הקריאות גם כאן מכיל כפילויות רבות, ולכן נוכל להשתמש שוב בטבלא.

נגדיר טבלא: בגודל $(n+1) \times (m+1)$, נמלא אותה בצורה אלכסונית או בצורת מרובע:

n	$\searrow$	$\searrow$				$\uparrow\longrightarrow$	$\uparrow\longrightarrow$	$\uparrow\longrightarrow$
$\searrow$	$\searrow$	$\searrow$	$\searrow$			$\uparrow\longrightarrow$	$\uparrow\longrightarrow$	$\uparrow\longrightarrow$
$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$		$\uparrow\longrightarrow$	$\uparrow\longrightarrow$	$\uparrow\longrightarrow$
$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\ldots$		
$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$		
$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	
0	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	m

כלומר בצורת המרובע אנו הולכים ימינה בשורות ועולים למעלה בעמודות בכל פעם. את החישוב עם האלכסונים ראינו בהרצאה. סך הכל $\mathcal{O}(n \cdot m)$. נחלץ את התוצאה מ-$M[n,m]$.

דוגמה. נניח כי $n = 4, m = 3$ עם $X = ABCD, Y = BDC$. נקבל את הטבלא:

D	4	0	$\downarrow 1$	$\swarrow 2$	$\leftarrow 2$
C	3	0	$\downarrow 1$	$\leftarrow 1$	$\swarrow 2$
B	2	0	$\swarrow 1$	$\leftarrow 1$	$\leftarrow 1$
A	1	0	$\leftarrow 0$	$\leftarrow 0$	$\leftarrow 0$
	0	0	0	0	0
		0	1	2	3
			B	D	C

איך אנו מסיקים את המחוזרת עצמה? בכל שלב נשמור את הצעד הקודם שהיינו בו ונחזור אחורה, נקבל (B,D). אם היינו בוחרים תא אחד למטה היינו מקבלים (B,C). זמן הריצה עדיין $\mathcal{O}(n\cdot m)$.

הוכחת נכונות

הוכחה: נוכיח באינדוקציה על $0\le i\le n$ ו-$0\le j\le m$ לפי סדר מילוי הטבלא כי בתא $M[i,j]$ אורך התמ"א של X^i, Y^j.

<u>בסיס</u>: עבור $i=0\vee j=0$ אחת המחרוזות ריקה ולכן התמ"א היחידה באורך 0.

<u>שלב</u>: נתבונן ב-i,j. נניח כי כל ה-"אינדקסים הקודמים" מולאו נכון. תהי $S=s_1,\ldots,s_r$ תמ"א של X^i,Y^j.

אם $x_i=y_j$ אזי $s_r=x_i$ אחרת $S'=s_1,\ldots,s_r,x_i$ היא גם תמ"א של X^i,Y^j והיא ארוכה יותר. במקרה זה $s_1,\ldots,s_{r-1}$ היא תמא של X^{i-1},Y^{j-1} ואכן ב-$M[i-1,j-1]$ כתוב $r-1$, מדרך פעולת האלגוריתם ב-$M[i,j]$ כתבנו $M[i-1,j-1]+1=r$.

אם $x_i\neq y_j$ אזי $s_r\neq x_i$ או $s_r\neq y_j$. במקרה זה S היא בהכרח תמ"א של X^i,Y^{j-1} או X^{i-1},Y^j ויתרה מכך היא הארוכה מבין ה-2. אנו מחשבים את $\max\{M[i-1,j],M[i,j-1]\}$, מהנחת האינדוקציה זה הערך הנכון. ■

9.3 שלבים לפתרון בעיה בתכנון דינאמי

באופן כללי, כאשר אנו נתקלים בבעיה ורוצים לפתור אותה באמצעות תכנון דינאמי נעקוב אחר הצעדים הבאים:

1. הגדרת תתי הבעיות.
2. כתיבת נוסחת רקורסיה.
3. הגדרת טבלא, סדר מילויה ואופן חילוץ הפתרון האופטימלי ממנה.
4. ניתוח זמן הריצה (לרוב מספר התאים בטבלא כפול הזמן למילוי כל תא).
5. הוכחת נכונות (חוקיות ואופטימליות) נוסחת הרקורסיה, לרוב באינדוקציה על סדר מילוי הטבלא.

הערה. כשנותנים לנו בעיה עם קונטקסט של תכנון דינאמי, מיד נבין שכדאי להשתמש בצעדים אלה, אך חשוב לזכור, בפועל, בעיות לא באות עם קונטקסט ועלינו להבין לבד מה הדרך הנכונה לפתור אותה. פעמים רבות הפרד ומשול זו דרך מועילה, ולכן אם נשתמש בה, נשקול אם להשתמש בצעדים אלה, או לפנות לדרכים אחרות.

10 בעיית כפל מטריצות

שתי מטריצות

שאלה כמה עולה לכפול שתי מטריצות מלבניות? תהי A מטריצה $n\times t$ ותהי B מטריצה $t\times m$ רוצים לחשב את $C_n=A_tB_m$. כמה כפלים של מספרים לוקח כדי לחשב את C?

לכל $1\le i\le n$ ולכל $1\le j\le m$ נחשב $C_{ij}=\sum\limits_{k=1}^{t}a_{ik}b_{kj}$ על ידי t כפלים. סך הכל כדי לחשב את כל C נצטרך $n\cdot t\cdot m$.

אם $n=m=t$ אזי דרושות $\Theta\left(n^3\right)$ פעולות כפל. יש מחקר לחישוב כפל מטריצות, כיום האלגוריתם הטוב ביותר עושה זאת ב-$\Theta\left(n^{2.37}\right)$, אנחנו לא מסוגלים להוכיח חסם תחתון, מעבר ל-n^2 שכן יש n^2 איברים. מטריצות עם הרבה אפסים, גם אותן חוקרים ויש דרכים יעילות יותר לחישוב הכפל שלהן.

10.1 הבעיה הכללית ופתרונה

נרצה לחשב את המטריצה $D = A \cdot B \cdot C$. ראשית אנו יודעים כי הכפל האסוציאטיבי $C = (A \cdot B) \cdot C = A \cdot (B \cdot C)$. אלגוריתמית, מתברר שזה לא אותו מחיר.

דוגמה. ניקח ${}_{10}D_{100} = {}_{10}A_{50} \cdot B_{20} \cdot C_{100}$. אזי האפשרות הראשונה היא $D = (A \cdot B)_{20} \cdot C_{100}$. חישוב $A \cdot B$ הוא $10 \cdot 50 \cdot 20 = 10^4$. לכן סך הכל $3 \cdot 10^4 = 10 \cdot 20 \cdot 100 + 10 \cdot 50 \cdot 20$.

בדרך השנייה אנו מחשבים $D ={}_{10} A_{50} \cdot (B \cdot C)_{100}$ וחישוב $B \cdot C$ דורש $50 \cdot 20 \cdot 100 = 10^5$, הרבה יותר מהסדר הקודם.

מכאן אנו מקבלים את הבעיה הבאה.

בעיה. בהנתן n מטריצות, מהו הסדר הטוב ביותר לחישוב המכפלה שלהן?

קלט $n+1$ מספרים טבעיים $p_0, p_1, \ldots, p_n$ מסמנים מימדים של n מטריצות $A_1, \ldots, A_n$ כאשר המטריצה A_i היא $p_{i-1} \times p_i$.

פלט סדר כפלי של מטריצות (חלוקת סוגריים) המשיג מחיר מינימלי לחישוב המכפלה $B = A_1 \cdot A_2 \cdot \ldots \cdot A_n$.

נציע כמה פתרונות:

- נאיבי - מעבר על כל האפשרויות, הגודל של מרחב הפתרונות החוקיים (דומה למספרי $catalan \sim 4^n$) יקר מדי.
- חמדני? מסתבר שלא.
- דינאמי - איך נחלק את הבעיה הגדולה לתת בעיות?
  - ◁ לפי המכפלה הראשונה שמבצעים.
  - ◁ לפי המכפלה האחרונה שמבצעים: עבור $1 \le k \le n-1$ נחשב $\underbrace{(A_{k+1} \cdot \ldots \cdot A_n)}_{D}\underbrace{(A_1 \cdot A_2 \cdot \ldots \cdot A_k)}_{C}$.

נסתכל על האפשרויות לחישוב הכפל לפי הפעולה הראשונה:

$$\begin{array}{cccc} A_1 & A_2 & A_3 & A_4 \\ (A_1 \cdot A_2) \cdot A_3 \cdot A_4 & A_1 (A_2 A_3) A_4 & A_1 A_2 (A_3 A_4) & \\ ((A_1 \cdot A_2) \cdot A_3) \cdot A_4 & (A_1 \cdot A_2)(A_3 \cdot A_4) & & \end{array}$$

לעומת זאת, אם נחשב מהכפל האחרון, נקבל

$$\begin{array}{cccc} A_1 & A_2 & A_3 & A_4 \\ A_1 (A_2 \cdot A_3 \cdot A_4) & (A_1 \cdot A_2) \cdot (A_3 \cdot A_4) & (A_1 \cdot A_2 \cdot A_3) \cdot A_4 & \\ A_1 \cdot (A_2 \cdot (A_3 \cdot A_4)) & A_1 \cdot ((A_2 \cdot A_3) \cdot A_4) & & \end{array}$$

כלומר תתי הבעיות מוגדרות בצורה ברורה, עבור A_i נסתכל על חלוקת הסוגריים מ-$A_i, \ldots, A_j$ עבור $i < j$.

למעשה, רק אחת יעילה. אחת תתן אלגוריתם יעיל והשנייה תתן המון תתי בעיות מיותרות. הדרך השנייה היא הטובה. אינטואיטיבית זה נובע מכך שהמטריצות בתתי הבעיות של הראשונה הן מטריצות כלשהן, שמייצרות עד הסוף מספר אקספוננציאלי של אפשרויות. נרצה להבין בדיוק למה.

נחלק את המטריצות ל-$\frac{n}{2}$ זוגות:

$$1,2 \;\; 2k-1, 2k, \quad , \ldots , \quad , n-1, n$$

עתה, נסתכל על פיצולים כשכל פעם בוחרים רק אחד מהזוגות ומכפילים אותו. בכל צעד נכפול רק כאלה. הפיצול הראשון הוא בגודל $\frac{n}{2}$. נסתכל מה קורה כאשר אנו בעומק $\frac{n}{4}$, אנו בוחרים למעשה $\frac{n}{4}$ זוגות מתוך $\frac{n}{2}$ הזוגות האלה ומכפיל אותם. לכל אחת מכפולות אלה נקבל תוצאה חדשה, לכן נקבל לפחות $\binom{\frac{n}{2}}{\frac{n}{4}} \sim \frac{2^{\frac{n}{2}}}{\sqrt{n}}$ תוצאות אפשריות לביצוע חלוקה זו.

החלוקה לפי פעולת הכפל האחרונה

שאלה מהן כל תת הבעיות שנפגוש?

הערה. נבחין כי אין לנו כאן עץ, שכן אנו חוזרים על בעיות שכבר פתרנו. למעשה אנו שמים חוצץ ברשימה באורך n עם n אפשרויות ולאחר מכן חוצצים משמאל עם n אפשרויות, ולכן סך הכל $\Theta\left(n^2\right)$ אפשרויות.

כל תת בעיות מהצורה הבאה: $A_i, A_{i+1}, \ldots, A_j$ עבור $1 \leq i \leq j \leq n$. כמה אפשרויות כאלה יש? צריך לבחור את כל הזוגות i, j כך ש-$i < j$, דהיינו $\binom{n}{2} = \Theta\left(n^2\right)$ אבל צריך להוסיף n למקרה בו $i = j$ לכן $\binom{n}{2} + n = \Theta\left(n^2\right)$.

סימון נסמן ב-$P\left[i, j\right]$ את המחיר האופטימלי לחישוב $A_i, \ldots, A_j$. בפרט $P\left[1, n\right]$ המחיר האופטימלי לחישוב $B = A_1 A_2 \ldots A_n$.

נרשום <u>נוסחת רקורסיה כללית</u>. עבור $\left(A_i \cdot \ldots \cdot A_k\right)\left(A_{k+1} \cdot \ldots \cdot A_j\right)$ מעקרון בלמן, מתקיים כי המחיר הוא חישוב המחיר לכל תת בעיות פלוס מחיר האיחוד שזה כפל המטריצות, על כן $P\left[i, k\right] + P\left[k, j\right] + p_{i-1} p_k p_j$. מכאן נסיק את הנוסחא הבאה:

$$P\left[i, j\right] = \begin{cases} i = j & 0 \\ i < j & \min_{i \leq k \leq j-1}\left\{P\left[i, k\right] + P\left[k+1, j\right] + p_{i-1} p_k p_j\right\} \end{cases}$$

אם כך נרשום את האלגוריתם הבא:

- נגדיר טבלא T בגודל $n \times n$ ונרשום בכל תא $T\left[i, j\right]$ לכל $1 \leq i, j \leq n$ את $P\left[i, j\right]$. בסופו של דבר נחזיר את $T\left[1, n\right]$.
  - ◁ כאשר $i > j$ נגדיר $T\left[i, j\right] = 0$.

דוגמה. נסתכל על 3×3: עם הגדלים $10, 50, 20, 100$: אנחנו הולכים לפי האלכסונים

3	$\star_2$	10^5	0
2	10^4	0	0
1	0	0	0
(j, i)	1	2	3

נחשב

$$\begin{aligned} T\left[1, 3\right] &= \min_{1 \leq k \leq 2}\left\{P\left[1, k\right] + P\left[k+1, 3\right] + 10 \cdot p_k \cdot 100\right\} \\ &= \min_{1 \leq k \leq 2}\left\{P\left[1, k\right] + P\left[k+1, 3\right] + 1000 \cdot p_k\right\} \\ &= \min\left\{0 + 10^5 + 5 \cdot 10^4, 10^4 + 0 + 2 \cdot 10^4\right\} \\ &= 3 \cdot 10^4 \end{aligned}$$

כדי לקבל גם את החלוקה עצמה, נשמור את הערך שבחרנו במעבר. זו רק דוגמא.

באופן כללי, נמלא את הטבלא כך:

$\star$	$\iddots$	$\iddots$	$\nearrow$	0
$\iddots$	$\iddots$	$\iddots$	0	
$\iddots$	$\nearrow$	$\iddots$		
$\nearrow$	0			
0				התא ה-(n)

עולים לפי אלכסונים. מכאן נוכל לרשום אלגוריתם <u>למילוי הטבלה</u>.

- נמלא $T[i,j]=0$ לכל $i>j$.
- נמלא את שאר הטבלה ב-n איטרציות.
- באיטרציה $0\le d\le n-1$ נמלא את הטבלה את התאים $T[i,j]$ עבורם $j-i=d$, לפי נוסחת הרקורסיה:

$$T[i,j]=\begin{cases} i=j & 0 \\ i<j & \min_{i\le k\le j-1}\left\{T[i,k]+T[k+1,j]+p_{i-1}p_kp_j\right\}\end{cases}$$

זמן ריצה

טענה. <u>זמן היצה</u> נטען כי מילוי כל תא בטבלא ניתן לעשות בזמן $\mathcal{O}(n)$.

הוכחה: לשם כך מספיק לוודא שבעת מילוי התא $T[i,j]$ עם $j-i=d$ התאים $T[i,k],T[k+1,j]$ לכל $i\le k\le j-1$ כבר מולאו באיטרציות קודמות. מתקיים מכיוון ש-$i\ge k$ כי $j-(k+1)<j-i\le d$. מכיוון ש-$j<k$ מתקיים כי $k-i<j-i=d$ כרצוי. ■ עתה, יש סך הכל $\mathcal{O}\left(n^2\right)$ תאים ב-T, ומילוי כל הטבלא T עולה $\mathcal{O}\left(n^3\right)$.

שחזור

נותר לנו להראות איך נוכל לשחזר את חלוקת הסוגריים.

כדי לחלץ את חלוקת הסוגריים האופטימלית נשמור בכל תא $T[i,j]$ עבור $i>j$ בעת מילויו את ערך ה-k המשיג את המינימום בנוסחת הרקורסיה, המראה את מיקום האופטימלי של פעולת ההכפלה האחרונה.

10.2 הוכחת נכונות

טענה. לכל $1\le i\le j\le n$ מתקיים כי $T[i,j]=P[i,j]$.

הוכחה: נוכיח באינדוקציה על $d=j-i$.

<u>בסיס</u>: $d=0$ כלומר $i=j$. במקרה זה $T[i,j]=0=P[i,j]$.

<u>שלב</u>: נניח עבור $0,\dots,d-1$ ונוכיח ל-d. לפי נוסחת הרקורסיה $d>0$, מתקיים כי

$$T[i,j]=\min_{i\le k\le j-1}\left\{T[i,k]+T[k+1,j]+p_{i-1}\cdot p_k\cdot p_j\right\}$$

ראינו כי לכל $i\le k\le j-1$ מתקיים כי $d>k-i$, $d>j-(k+1)$ ולכן לפי הנחת האינדוקציה נקבל כי

$$T[i,j]=\min_{i\le k\le j-1}\left\{P[i,k]+P[k+1,j]+p_{i-1}\cdot p_k\cdot p_j\right\}$$

כרצוי. ■

11 בעיית התרמיל השלם

רקע

11.0.0.1 סיפור מסגרת אנחנו בחנות אופנועים. תסיקו את ההמשך לבד, בדידה.

זו הבעיה הדינאמית האחרונה שלנו. יהיה מעניין, כי זו בעיה NP קשה, דהיינו אין אלגוריתם יעיל שפותר אותה.

קלט המשקל המירבי של התרמיל W ו-n וזוגות של מספרים $(v_1, w_1), (v_2, w_2), \ldots, (v_n, w_n)$ כאשר v_i הערך של הפריט ה-i ו-w_i משקל הפריט ה-i. הפריטים אינם ניתנים לחלוקה.

פלט תת קבוצה $S \subseteq [n]$ כך ש-$\sum\limits_{i \in S} w_i \leq W$ וכך ש-$\sum\limits_{i \in S} v_i$ מקסימלי.

נציע פתרון נאיבי. נעבור על כל תת הקבוצות של $[n]$ וזה עולה $\Omega(2^n)$.

האם יש פתרון טוב יותר?

- חמדני? מטרואידי: נמיין את הפריטים לפי ערכם בסדר יורד ובכל שלב נכניס הפריט היקר ביותר לתרמיל.
- ערך סגולי: נמיין את הערכים לפי ערכם הסגולי $r_i = \frac{v_i}{w_i}$ בסדר יורד ובכל שלב נכניס הפריט היקר ביותר בערכו הסגולי הנכנס לתרמיל.

ננסה למצוא דוגמאות נגדיות לשני הפתרונות.

- מטרואידי: נבחר ערך בעל משקל גדול יותר משאר המשקלים בעל ערך גדול מכולם כך שלאחר שנכניס אותו לא נוכל להכניס עוד ערכים, אבל אם נכניס את שני הערכים הבאים נקבל ערך טוב יותר.
- ערך סגולי: נבחר ערכים עם משקלים קטנים שימלאו את התרמיל אבל יהיו בעל ערך קטן יחסית, וערך שיהיה גדול מהסכום שלהם אבל בעל משקל התרמיל. דהיינו $W = 10$, $(3, 10), (1, 2), (1, 1)$. לפי האלגוריתם אנו נבחר את $(1, 2), (1, 1)$ למרות שיכולנו לבחור את $(3, 10)$.
- נוכל לתת דוגמא נגדית שסותרת את שניהם: $W = 50, n = 3, (150, 30), (100, 25), (100, 25)$. לפי שני העקרונות אנו נכניס תחילה את הפריט הראשון ולא נכניס שום דבר יותר, נרוויח 150. למרות שהפתרון האופטימלי הוא להכניס את שני הפריטים האחרים.

שאלה כיצד נבנה אלגוריתם דינאמי לפתרון הבעיה?

נציע כמה פתרונות:

- נפתור את בעיית התרמיל עם הפריטים $\{1, \ldots, i\}$ ומשקל מירבי $0 \leq u \leq W$, עם $i < n$.

  בעיה. אין התייחסות לפריט ה-n.

- נסדר את הפריטים בסדר כלשהו. ההחלטה לגבי הפריט הראשון מפרקת את הבעיה לשתי בעיות קטנות יותר.

נרשום דיאגרמה למה שקורה:

$$\begin{array}{ccccc} & \swarrow & ([n], W) & \searrow & \\ \swarrow & +\{2, \ldots, n\}, W - w_1 - & \swarrow & +\{2, \ldots, n\}, W - & \searrow \\ \{3, \ldots, n\}, W - w_1 - w_2 & & \{3, \ldots, n\}, W - w_2 & & \{3, \ldots, n\}, W \end{array}$$

כיצד נתאר את תתי הבעיות?

נתאר את תת הבעיות שנפגוש. הן מהצורה הבאה: בעיית התרמיל השלם שהפריטים בה הם $\{i, \ldots, n\}$ כאשר $1 \leq i \leq n$ ומשקל מירבי u כאשר $0 \leq u \leq W$. נראה שמספר המשקלים האפשריים u יהיה גדול מדי (אקספוננציאלי ב-n) ולכן האלגוריתם לא יהיה יעיל.

סימון נסמן ב-$K\left[i,u\right]$ את הערך האופטימלי של בעיית התרמיל השלם עם פריטים $\{i,\ldots,n\}$ ומשקל מירבי u.

הערך של הבעיה כולה הוא $K\left[1,W\right]$. נוסחת הרקורסיה המתארת את הפיצול הראשון:

$$K\left[1,W\right]=\max\left\{K\left[2,W-w_1\right]+v_1,K\left[2,W\right]\right\}$$

נרשום עתה נוסחה כללית:

$$K\left[i,u\right]=\begin{cases} i=n,w_n>u: & 0\\ i=n,w_n\le u & v_n\\ i<n,w_i>u & K\left[i+1,u\right]\\ i<n,w_i\le u & \max\left\{\underbrace{K\left[i+1,W-w_1\right]+v_1}_{\text{נכניס}},\underbrace{K\left[i+1,W\right]}_{\text{לא נכניס}}\right\}\end{cases}$$

בנינו אם כך נוסחת רקורסיה. נותר לנו החלק האלגוריתמי.

11.1 גדילה אקספוננציאלית

נבנה טבלה T ונרשום בתא $T\left[i,u\right]$ נמלא את הערך $K\left[i,u\right]$, נחזיר $T\left[1,W\right]$. זה נראה כך:

	n			
$T:$				
i	1			
		u		

אילו ערכים של u יכולים להופיע בתת הבעיה? נקבע $1\le i\le n$ ונתבונן בבעיה $K\left[i,u\right]$, עם הערכים שהכנסנו$R\subseteq\{1,\ldots,i-1\}$.
מהם הערכי u האפשריים?

כל ערך כזה הוא מהצורה הבאה $W-\sum\limits_{i\in R}w_i$ עבור $R\subseteq\{1,\ldots,i-1\}$. כל המשקלים האפשריים בשלב הזה שייכים לקבוצה

$$\left\{W-\sum_{i\in R}w_i,R\subseteq\{1,\ldots,i-1\}\right\}$$

יש כאן 2^{i-1} אפשרויות, אבל מי אמר שהן שונות? נוכל לקחת מספרים ממשיים, כל מיני $\sqrt{\pi},e,\pi$ שאין סיבה שהסכומים שלהם שווים. אבל יותר פשוט, נוכל לבחור חזקות של 2, ונקבל ייצוג בינארי שידוע שהוא ייחודי. נגדיר אם כך $w_i=2^i$ עבור $1\le i\le n$ אזי עבור $R_1\neq R_2$ נקבל כי $\sum\limits_{i\in R_2}w_i\neq\sum\limits_{i\in R_1}w_i$ ולכן נקבל 2^{i-1} אפשרויות שונות ל-u בשלב זה.

זה בעייתי מאוד. עבור $i=\frac{n}{2}+1$ נקבל $2^{\frac{n}{2}}$ תת בעיות שונות (ערכים שונים של u) שזה יקר מדי. נצטרך הנחות מקלות.

- W הוא מספר טבעי וכל המשקלים של הפריטים גם הם מספרים טבעיים.

◁ במקרה זה המשקלים u בתת הבעיות הם מהצורה הבאה: $0\le u\le W$ ולכן נקבל טבלא עם n שורות ו-$W+1$ עמודות, כלומר טבלא בגודל $O\left(n\cdot W\right)$ שזה טוב יותר, אך אם W גדול מדי זה בעייתי.

- זו למעשה הבעיה בכל שמדובר בבעיה NP קשה.

11.2 האלגוריתם

נבנה טבלא T עם n שורות ו-$W+1$ עמודות כאשר נמלא בתא $T[i,u]$ את הערך $K[i,u]$ לכל $1 \le i \le u$ ו-$0 \le u \le W$.

מילוי הטבלא. נבחין כי ככל ש-i גדל, מספר הפריטים בבעיה קטן. לכן נתחיל למלא את הטבלא מהשורה ה-n-ית. בשורה זו רשומים רק שני איברים אפשריים, או 0 או v_n. הסף בטבלא הוא עד w_n, אם $u < w_n$ נקבל 0 והחל משם v_n. במקרה שאנו נמצאים בתא כללי, אנחנו צריכים לבחור במקסימום בין הבחירה בערך הנוכחי, לבין מעבר לאחד הבא. על כן היא תראה כך:

	n	0	$\dots$	0	v_n	$\dots$	v_n
			$+v_i$	$no\ v_i$			
T_i				$\nwarrow\uparrow$			
							$\star$
	0	$u-w_1$					W

סך הכל, אנו ממלאים את הטבלא שורה אחר שורה מהשורה n לשורה הראשונה. את השורה ה-n נמלא לפי תנאי השפה המתאים של הרקורסיה. את השורות הבאות נמלא ב-$n-1$ איטרציות כאשר באיטרציה $1 \le j \le n-1$ נמלא את השורה $i = n-j$ באופן הבא:

$$T[i,u] = \begin{cases} T[i+1,u] & w_i > u \\ \max\{T[i+1,u-w_i]+v_i, T[i+1,u]\} & w_i \le u \end{cases}$$

לאחר מילוי הטבלא נחזיר את $T[1,W]$. כדי לחלץ את הפתרון - אלה פרטים נכניס לתרמיל - נזכור בעת מילוי כל תא $T[i,u]$ את ההחלטה הנכונה (להכניס או לא להכניס) את הפריט ה-i, המשיגה את המקסימום בנוסחת הרקורסיה.

כדי לחשב את זמן הריצה, נבחין כי בעת מילוי כל תא צריך לבדוק מה רשום בשני תאים בשורה מעליו, שאותם מילאנו באיטרציה הקודמת. לכן זה עולה $O(1)$, וסך כל זמן הריצה הוא $O(n \cdot W)$.

שאלה האם זמן הריצה $O(n \cdot W)$ יעיל?

התשובה טמונה בערכו של W. אם W פולינומי ב-n, זמן הריצה הוא פולינומי ב-n וזה טוב. אחרת, אם למשל $W = 3^n$, זמן הריצה יהיה גדול יותר מזה של פתרון נאיבי. האמנם? נבחין כי גדול הקלט הינו $O(n \log W)$ שכן יש לנו $n+1$ משקלים שהם לכל היותר W. על כן זמן הריצה הוא $\Omega(n \cdot W)$, הקושי פה הוא שגודל הקלט הוא לוגריתמי ב-W ושווה ל-$n \log W$ וזמן הריצה תלוי לינארית ב-W ושווה ל-$n \cdot W$. כאן נכנסת העובדה שמדובר בבעיה NP קשה.

12 תרגול 5 - אלגוריתמים דינאמיים, בעיית מסילת הרכבת ובעיית APSP

12.1 בעיית מסילת הרכבת

12.1.0.1 סיפור רקע הגענו לרציף $9\frac{3}{4}$ ואנו רוצים להרכיב מסילת רכבת בעלות מינימלית כדי שנוכל לשטות במוגלגים שאכן מדובר במסילת רכבת רגילה, כאלה אנחנו, מתוחכמים. אממה, וולדמורט המרושע החליט להטיל קשיים על המשימה, הוא הטיל כישוף שגרם לכך שכל חלק שאנו בוחרים להתחלת הצעד הבא במסלול חייב להסתיים עם חלק ייעודי, ולשני החלקים ביחד יש מחיר ואורך מסוים. כיצד נבנה את המסילה בצורה אופטימלית ונתגבר על המכשף הרשע?

קלט $L \in \mathbb{N}$ אורך המסילה, $\{1,\dots,K\}$ סוגי חיבורים, N רביעיות מהצורה $\{s_i,e_i,d_i,p_i\}$ כאשר $s_i,e_i \in [k]$, s_i החיבור בהתחלה ו-e_i החיבור בסוף, d_i אורך החלק, p_i מחיר החלק.

פלט המחיר המינימלי להרכבת מסילה חוקית באורך L, כאשר מסילה חוקית היא מסילה שבה חלק i מופיע מימין לחלק j אם"ם $s_i = e_j$.

דוגמה. $L=3$, $k=4$ עם החיבורים $\{(\ni,\in),(o,o),(>,<),(],[)\}$ והחלקים
$\{(\ni,o,1,30),(>,\in,1,10),(],[,1,10),(o,o,2,40),(],\in,3,100)\}$ והחלקים יראו כך

$$\begin{array}{ccccccccc} & 1 & & & 1 & & & & \\ \ni & - & o &] & - & [& & & \\ & 30 & & & 10 & & & & \\ & 1 & & & 2 & & & 3 & \\ > & - & \in & o & - & o &] & - & \in \\ & 10 & & & 40 & & & 100 & \end{array}$$

לא אכפת לנו כל כך מהחיבורים בתחילת וסוף המסילה, אבל מה שכן שמנה אלה החיבורים באמצע, שכן] לא יכול להתחבר לאף חלק מלבד [.

כיצד נפתור בעיה? נבחין כי בבעיות כאלה ננסה לזהות את תת הבעיה, שנפתור בצורה רקורסיבית. למשל, נתחיל מהסוף ונציב חלק בקצה. גישה נפוצה לעיצוב אלגוריתם דינאמי היא להסתכל על הצעד האחרון בפתרון אופטימלי כלשהו ולנסות להבין האם כשמורידים צעד זה נשארים עם פתרון אופטימלי עבור משימה כלשהי, דהיינו האם עקרון בלמן מתקיים. אם כן, המשימות האלו מגדירות לנו את תתי הבעיות. הדרך בה מחליטים מה הוא הצעד האחרון על סמך אוסף פתרונות תתי הבעיות תגדיר עבורנו את נוסחת הרקורסיה.

- <u>תתי הבעיות</u> לכל $0 \le l \le L$ ולכל $1 \le k \le K$ מה המחיר המינימלי של מסילה באורך l שמסתיימת בחיבור k?

⊲ נסיק את הפתרון לבעיה שלנו על ידי בחירת המינימום מבין ה-k, כלומר את ה-k שנותן את התוצאה הטובה ביותר.

- <u>נוסחת הרקורסיה</u> נסמן ב-$f(l,k)$ את המחיר המינימלי לבניית מסילה באורך l שמסתיימת בחלק ה-k. נבחין כי

$$f(l,k)=\begin{cases} 0 & l=0 \\ \min\limits_{\substack{i\in[N]\\ e_i=k\\ l-d_i>0}} \{p_i+f(l-d_i,e_i)\} & \exists i\in[N]: e_i=k, l-d_i>0 \\ \min\emptyset=\infty & otherwise \end{cases}$$

- <u>הגדרת הטבלה</u>: בגודל $K\times(L+1)$. נמלא אותה מ-$l=0$ עד $l=L$ ונמלא את השורה המתאימה באיזה סדר שנרצה, שכן מסתמכים אך ורק על השורה התחתונה, ולכן אין משמעות לסדר.

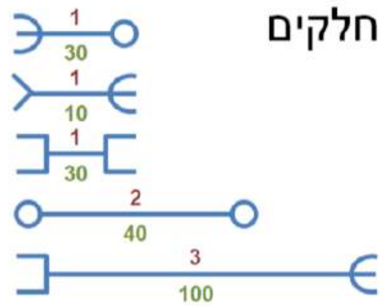
- <u>סיבוכיות</u>: הטבלה היא בגודל $O(K\cdot L)$ ואנו בכל צעד מבצעים פעולה שעולה $O(N)$ ולכן סך הכל $O(N\cdot K\cdot L)$.

דוגמאת הרצה

נחזור לדוגמא שלנו:

3	100	70	∞	90
2	∞	40	∞	60
1	10	30	∞	30
0	0	0	0	0
	1	2	3	4
	$\ni$	o	$>$	$]$

טבלה 1: פלט האלגוריתם $f\left(l,k\right)$



איור 2: דוגמא לנתוני הבעיה

על כן נקבל את הטבלה הבאה:

כדי לשחזר את המסלול נוכל לשמור פוינטר שיצביע בכל בחירת מינימום על החלק הבא שלקחנו ולעדכן אותי להיות האחד הבא, ככה שנשמור פוינטר בסיס. נקבל רשימה מקושרת של חלקים וממנה נסיק את המסילה.

12.2 בעיית APSP ואלגוריתם Floyd-Warshall

זו בעיה ידועה שראינו בדאסט. נתעלם מכך בצורה מופגנת.

קלט גרף מכוון $G=\langle V,E\rangle$, פונקציית משקל $w:E\to\mathbb{R}$ לאו דווקא חיובית. נסמן $|V|=n$.

פלט מטריצה בגודל $|V|\times|V|$ כך שבתא i,j כתוב משקל מינימלי של מסלול מ-i ל-j.

הנחה אין בגרף מעגל שסכום הקשתות שלו שלילי, אחרת הבעיה לא מוגדרת היטב (כי אז ניתן לעבור במעגל אינסוף פעמים ולשפר את המשקל של המסלול).

הצעה. אנו רוצים למצוא את המסלולים הקצרים ביותר בין כל קודקודי גרף ממושקל חסר מעגלים שליליים. האינטואיציה ההתחלתית, היא עבור זוג (v,w) לחשב את כל המסלולים הקצרים ביותר לכל אחד מהקודקדים האחרים ואז לבחור את המינימום בין חיבורי שני המסלולים דהיינו $v\to p\to w$.

מה הבעיה? הרקורסיה אינסופית! כי לא הקטנו כלל את הבעיה. אם כך, מה נעשה? צריך לחשוב על טקטיקה מעט שונה. כשננסה להגדיר את נוסחת הרקורסיה ניתקל בבעיה - מה הן תתי הבעיות הקטנות יותר ומה הן תתי הבעיות הגדולות יותר? כמו־כן, בשום דרך בה נגדיר את נוסחת הרקורסיה לא נצליח למלא את הטבלה ביעילות.

מסקנה. אנו חייבים להוסיף עוד פרמטר לבעיה שיעזור לקבוע את 'גודלה' של כל תת־בעיה ואת סדר חישוב הפתרון.

ננסה דרך אחרת:

- <u>תתי בעיות:</u> לכל i,j,k נמצא את משקל המסלול המינימלי בין i ל-j שמשתמש לכל היותר ב-k צלעות.

- נוסחת הרקורסיה: נגדיר $f(i,j,0)=\begin{cases}0 & i=j\\ \infty & i\neq j\end{cases}$. עתה, לכל $(i,j)\notin E$ נגדיר $w(i,j)=\infty$ וגם נגדיר $f(i,j,k)=\min\limits_{l\in V}\{f(i,l,k-1)+w(l,j)\}$.
  - ◁ ניתן להוכיח שהאלגוריתם עובד כמו שצריך.
- זמן ריצה: יש לנו טבלא בגודל $|V|^3$ כאשר בכל צעד אנו מבצעים $O(|V|)$ עבורה לכן סך הכל $O\left(|V|^4\right)$ עבודה סך הכל.

זה עובד, אבל יקר. האם יש דרך טובה יותר? האלגוריתם של פלוייד וורשל.

12.2.1 Floyd-Warshall

החידוש של Floyd-Warshall הוא השינוי בתפיסה של הקטנת תתי הבעיות.

- תתי בעיות: לכל i,j,k נמצא את משקל המסלול המינימלי בין i ל-j שמשתמש רק בקודקודים $\{1,\ldots,k\}$ כקודקודי ביניים.
- נוסחת הרקורסיה: $f(i,j,0)=\begin{cases}0 & i=j\\ w(i,j) & (i,j)\in E\\ \infty & otherwise\end{cases}$. לכל $k>0$ נגדיר

$$\boxed{f(i,j,k)=\min\{f(i,j,k-1),f(i,k,k-1)+f(k,j,k-1)\}}$$

דהיינו המסלול המינימלי עם הקודקודים $\{1,\ldots,k\}$ הוא המינימום בין המסלולים שמשתמשים ב-k לבין המסלולים שלא משתמשים בו.

- מילוי הטבלא: נגדיר טבלא בגודל $N\times N\times(N+1)$ כאשר ה-$+1$ הוא למקרה ש-$N=0$. נמלא עבור $k=0$ עד $k=n$, כאשר מכיוון שכל התוצאות מסתמכות רק על שורות קודמות, נמלא את הטבלא המתאימה ל-k באיזה סדר שנרצה.
  - ◁ נחזיר את $M[\star,\star,n]$.
  - ◁ כדי להחזיר את המסלול עצמו נוכל לשמור מידע נוסף בהתאם.
- זמן הריצה: יש לנו $|V|^3$ תאים בטבלא, מילוי כל תא הוא $O(1)$ ולכן סך הכל $O\left(|V|^3\right)$.

12.3 תת סדרה עוקבת עם סכום מקסימלי

12.3.0.1 סיפור רקע נתונה לנו סדרה של מספרים שלמים ואנו רוצים לחשב תת סדרה עוקבת (באינדקסים, לא בערכים) עם סכום מקסימלי.

קלט $a_1,\ldots,a_n\in\mathbb{Z}$.

פלט $i\leq j\in[n]$ כך ש-$\sum\limits_{k=i}^{j}a_i$ מקסימלי.

דוגמה. עבור $\{-1,5,3,-2,4,-4,-1,3,2\}$ הסדרה האופטימלית היא $-1,5,3,-2,4$.

נציע פתרון נאיבי - נעבור על כל האפשרויות - n^2 אופציות נסכום את כולן ונבחר מינימום, n^3 סך הכל. פולינומיאלי אבל לא יעיל מספיק.

האם יש פתרון טוב יותר?

נוכל לשמור בטבלא את תת הסדרה שמתחילה ב-i ונגמרת ב-j, וכך נחסוך חישובים חזרתיים. כל סדרה תסתמך על בעיות קודמות ותבחר את המינימום בין מספר קבוע של אפשרויות, שכן $f(i,j)=\min\{a_i+f(i+1,j),f(i+1,j)\}$, ולכן נקבל $O\left(n^2\right)$. טוב יותר!

האם יש דרך טובה יותר? נשאל, מה התת סדרה הכי טובה שמסתיימת ב-4? אם נדע אותה, נוכל להסתמך על רציפותה ולבחור או את האיבר 4, או 4 יחד עם הסדרה הקודמת.

- תתי בעיות: לכל i, מה הפתרון האופטימלי שמסתיים ב-i? כלומר, לוקח את a_i ולא ממשיך.
- רקורסיה: $f(i) = \begin{cases} a_i & i = 1 \\ \max\{a_i, a_i + f(i-1)\} & i > 1 \end{cases}$ דהיינו $f(i) = \begin{cases} a_i & i = 1 \\ a_i + \max\{0, f(i-1)\} & i > 1 \end{cases}$.
- הגדרת טבלה: נגדיר טבלא בגודל n, נמלא מ-$i = 1$ עד n, ונחזיר את $\max\left\{\max_{1\le i\le n}\{f(i)\}, 0\right\}$.
- חילוץ הפתרון: היינו יכולים לשמור שני אידקסים s, e, כאשר s יסמן את תחילת הסדרה ו-e את סופה. בכל שלב, אם מצאנו פתרון טוב יותר שנגמר במקום אחר, נעדכן את e ונעדכן את s כך שיתחיל במקום בו מתחיל הפתרון החדש - ידוע כי הוא מסתמך על בעיות קודמות. או שהיינו בוחרים לשמור לכל תא את ההתחלה שלו ולהחזיר את ההתחלה והאינדקס של הטוב ביותר כי האינדקס שלו הוא הסוף.

שאלה האם אפשר לעשות זאת בצורה יעילה יותר מבחינת זכרון? דהיינו ללא טבלא שלמה, אלא עם זכרון קבוע?

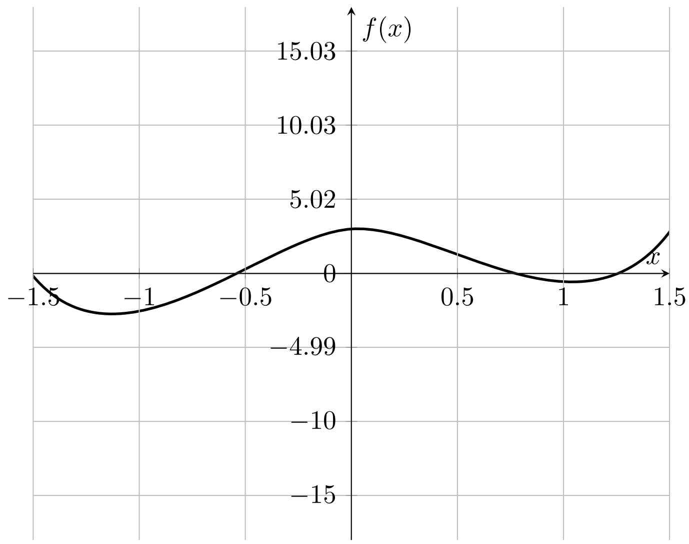
נבחין כי בכל שלב אנחנו יכולים לשמור אינדקס המסמל את האיבר המקסימלי שמצאנו עד כה, ערך שיסמן את הערך של החישוב הקודם, וערך שיסמן את החישוב הנוכחי. בכל שלב נעדכן את הערך הקודם להיות הנוכחי והערך הנוכחי להיות $f(i)$. נשמור בערך המקסימלי את המקסימום מבין שלושת האיברים - הקודם, הנוכחי והמקסימום הנוכחי. כדי להחזיר את המסלול נשמור לכל ערך את ההתחלה שלו והאינדקס שלו. קבוע סך הכל.

חלק IV

אלגוריתמי קירוב

מבוא

חלק גדול מבעיות אלגוריתמיות מעניינת כנראה לא ניתנות לפתרון יעיל. נחפש פתרונות מקורבים לבעיות כאלה. מה הכוונה בפתרון מקורב? כשדיברנו על אלגוריתמים חמדניים הגדרנו פונקציית משקל לבעיה $f:\mathcal{S}\to\mathbb{R}$ שלקחה פתרון ממרחב הפתרונות $\mathcal{S}$ והחזירה ערך ממשי שייצג "כמה טוב" הפתרון. לפונקציה זו יש ערכי מינימום ומקסימום, והם יכולים להיות מקומיים או גלובליים, למשל בפונקציה הבאה:



איור 3: פונקציה בעלת נקודת מינימום מקומית וגלובלית

במקום לבחור במינימום הגלובלי, נוכל לבחור בנקודה קרובה למינימום הלוקלי, שבמקרים מסוימים יהיה מספיק טוב עבורנו.

העולם של אלגוריתמי קירוב הוא מאוד מגוון, מלא בבעיות קשות ועשיר בכלים לפתרון בעיות. כמו בפרקים קודמים, נעמיק בנושא באמצעות דוגמאות מייצגות.

13 חלוקת משימות בין מכונות Load Balancing

זו בעיה מאוד חשובה, נתונים לנו למשל מעבדי מחשב ונרצה להשתמש בהם בצורה שווה מבלי להעמיס על אחד מהם בביצוע משימות מסוימות. נרצה לתת פתרון מקורב לבעיה.

קלט k - מספר של מכונות זרות ו-n מספרים חיוביים $t_1,\dots,t_n$ המסמנים זמני ריצה של n משימות.

פלט חלוקה מאוזנת כמה שאפשר הממזערת את זמן העבודה של המכונה העמוסה ביותר של המשימות בין k המכונות.

דוגמה. $k=2, n=4$ עם $\begin{matrix} t_4 & t_3 & t_2 & t_1 \\ 2 & \frac{1}{2} & \frac{1}{2} & 1 \end{matrix}$. נטען כי החלוקה המאוזנת ביותר היא $\left(\frac{1}{2},\frac{1}{2},1\to 1\right),(2)\,(2\to 2)$ עם משקל $q\left(S^\star\right)=2$.
איך אנו יודעים שהוא אופטימלי? יש כאן משימה אחת שעולה 2, ולכן לא משנה מה נעשה תמיד נצטרך לשלם לפחות 2 עבור מכונה אחת.

זו הייתה רק טעימה. אבל ממה שכבר למדנו, איזה אלגוריתם חמדן מתאים לפתרון הבעיה? נשלח אל המכונה הפנויה ביותר את המשימה הקרובה ביותר שראינו. בדוגמא זו, היינו מקבלים $(2,1\to 1)\,,\left(\frac{1}{2},\frac{1}{2}\to 1\right)$ עם $q\left(S\right)=3$ פחות טוב מהאופטימלי, אבל

אנו נרצה למדוד את היחס בין $\frac{q(S)}{q(S^\star)}$. אנו ננתח אלגוריתם זה ונראה שהוא עובד לא רע בכלל ביחס לאלגוריתם אופטימלי. כמו כל בעיה, נגדיר את המרחב שלנו, את המשקל והמטרה.

- מרחב הפתרונות: החוקיים לבעיה זו הוא $\mathcal{S} = \{S : [n] \to [k]\}$ ונבחין כי $|\mathcal{S}| = k^n$, לכן הפתרון הנאיבי של מעבר על כל האפשרויות הינו קשה.
- משקל: עבור $1 \le j \le k$ נגדיר $T_j(S)$ להיות זמן הריצה של המכונה ה-j לפי פתרון S כך ש-

$$T_j(S) = \sum_{\substack{1 \le i \le n \\ S(i) = j}} t_i$$

ונגדיר $q(S) = \max_{1 \le j \le k} T_j(S)$.

- מטרה: אנו נרצה להחזיר $S^\star \in \mathcal{S}$ כך ש-$q(S^\star) = \min_{s \in \mathcal{S}} q(S)$.

הערה. מהי חלוקה מאוזנת? זה תלוי באלגוריתם, במקרה שלנו בחירה חמדנית של כל מכונה.

הערה. האם המשימות ידועות מראש? או שהן מופיעות במהלך הריצה? זה תלוי. אם אנו מערכת ההפעלה, משימות יופיעו במהלך הריצה, ולכן נרצה לתת מענה למקרה זה. באופן כללי, אלגוריתמים שיודעים את הקלט "מראש", נקראים אלגוריתמים offline, בעוד אלגוריתמים שמקבלים אותה במהלך ריצתם, כמו מערכת ההפעלה, נקראים אלגוריתמי $online$. המטרה באלגוריתמים אלה היא למצוא את הפתרון הטוב ביותר. כיצד מוצאים? למה משווים? אנו משווים לפתרון ה-$offline$ וביחס אליו יודעים כמה אנחנו טובים. אנחנו נפתור את הבעיה כ-$online$.

אנו הצענו אלגוריתם חמדני באופן הבא. נחלק את המשימות בסדר הגעתן, כאשר המשימה מגיעה, נשלח אותה למכונה הכי פחות עמוסה ברגע זה. נרצה להוכיח שהוא מקרב את הבעיה. אבל עולה הבעיה. מה משמעות המושג מקרב?

נזכור כי בבעית אופטימיזציה נתון מרחב פתרונות חוקיים $\mathcal{S}$ לבעיה אלגוריתמית נתונה ופונקציית איכות $q : \mathcal{S} \to \mathbb{R}_+$. בבעיית מקסימיזציה נחפש $S^\star \in \mathcal{S}$ כך ש-$q(S^\star) = \max_{S \in \mathcal{S}} q(S)$ ובבעיית מינימיזציה $q(S^\star) = \min_{S \in \mathcal{S}} q(S)$.

הגדרה. יהי $C \ge 1$. נאמר כי האלגוריתם הינו C מקרב לבעיית מקסימיזציה נתונה עם פתרון אופטימלי $S^\star$, אם האלגוריתם מחזיר פתרון S כך ש-$(1 \le) \frac{q(S^\star)}{q(S)} \le C$.

הגדרה. יהי $C \ge 1$. נאמר כי האלגוריתם הינו C מקרב לבעיית מינימיזציה נתונה עם פתרון אופטימלי $S^\star$, אם האלגוריתם מחזיר פתרון S כך ש-$(1 \le) \frac{q(S)}{q(S^\star)} \le C$.

הערה. במקרה של מינימיזציה אנו יודעים כי $q(S^\star) \le q(S)$ ולכן $q(S^\star)$ במונה כדי שנקבל מנה גדולה שווה מ-1. במקרה של מקסימיזציה זה הפוך.

אנו נחפש אלגוריתמים כאלה, אבל יעילים.

מסקנה. ככל ש-α גדול יותר כך האלגוריתם טוב יותר. עבור $\alpha = 1$ קיבלנו אלגוריתם אופטימלי.

השערה. אם α קטן יותר, זמן הריצה גדול יותר.

משפט. האלגוריתם החמדני שתיארנו הינו אלגוריתם $\left(2 - \frac{1}{k}\right)$ מקרב לבעיה.

לפני הוכחת המשפט, נוכיח למות על איכות הפתרון האופטימלי.

למה. (1) נסמן ב-t_{max} את זמן הריצה של המשימה הארוכה ביותר, אזי $q(S^\star) \ge t_{max}$.

הוכחה: מושאר כתרגיל לקוראת הנאמנה. ∎

למה. (2) מתקיים כי $q(S^\star) \ge \frac{1}{k} \sum_{i=1}^{n} t_i$.

הוכחה: מההגדרה, ומכך שהמקסימום של קבוצה גדול מהממוצע שלה,

$$q\left(S^{\star}\right)=\max_{1\leq j\leq k}T_{j}\left(S^{\star}\right)\geq\frac{1}{k}\sum T_{j}\left(S^{\star}\right)=\frac{1}{k}\sum_{j=1}^{k}\sum_{\substack{1\leq i\leq n\\ S^{\star}(i)=j}}t_{i}=\frac{1}{k}\sum_{i=1}^{n}t_{i}$$

כרצוי. ■ בהוכחת המשפט, נכנס פנימה, אך תוך האלגוריתם, ונשתמש בלמות. האינטואיציה היא שעבור המכונה העמוסה ביותר נוכל להסתכל על המשימה האחרונה שהיא קיבלה, ולהסתכל על הרגע לפני שהיא קיבלה אותה. ברגע זה מאופן פעולת האלגוריתם החמדן, היא הייתה המכונה הפנויה ביותר, מישהי עלולה לטעון כי המשימה החדשה ארוכה מאוד, אבל בכל מקרה צריך לטפל בה ולכן השמתה במכונה הפנויה ביותר שומרת על אופטימליות. נפרמל את מה שאמרנו בהוכחה.

הוכחה: (המשפט) יהי $S\in\mathcal{S}$ הפתרון החמדן ויהי $S^{\star}\in\mathcal{S}$ פתרון אופטימלי. יהי $1\leq j_0\leq k$ האינדקס של המכונה העמוסה ביותר, כלומר $q\left(S\right)=T_{j_0}\left(S\right)$. יהי $1\leq l\leq n$ האינדקס של המשימה האחרונה שנשלחה למכונה j_0.

לפי דרך פעולתו של האלגוריתם החמדן, המכונה j_0 היא המכונה הכי פחות עמוסה אחרי החלוקה של $l-1$ המשימות הראשונות.

עבור $1\leq j\leq k$ נסמן ב-$F_j\left(S\right)$ את זמן הריצה של המכונה ה-j לאחר החלוקה של $l-1$ המשימות הראשונות, דהיינו $F_j\left(S\right)=\sum_{\substack{1\leq i\leq l-1\\ S(i)=j}}t_i$ אזי $F_{j_0}\left(S\right)=\min_{1\leq j\leq k}F_j\left(S\right)$. אזי מכך שהמינימום של k מספרים קטן מהממוצע שלהם נקבל,

$$\begin{aligned}
q\left(S\right)=T_{j_0}\left(S\right)=t_l+F_{j_0}\left(S\right)=t_l+\min_{1\leq j\leq k}F_j\left(S\right)&\leq t_l+\frac{1}{k}\sum_{j=1}^{k}F_j\left(S\right)\\
&=t_l+\frac{1}{k}\sum_{j=1}^{k}\sum_{\substack{1\leq i\leq l-1\\ S(i)=j}}t_i=t_l+\frac{1}{k}\sum_{i=1}^{l-1}t_i\\
&=\left(1-\frac{1}{k}\right)t_l+\frac{1}{k}\sum_{i=1}^{l}t_i\\
&\leq\left(1-\frac{1}{k}\right)t_{max}+\frac{1}{k}\sum_{i=1}^{n}t_i\\
&\overset{\text{Lem.}}{\leq}\left(1-\frac{1}{k}\right)q\left(S^{\star}\right)+q\left(S^{\star}\right)\\
&=\left(2-\frac{1}{k}\right)q\left(S^{\star}\right)
\end{aligned}$$

כרצוי. ■ זה היה מאוד פשוט, אבל גם מאוד מפתיע, שאנחנו יכולים לקבל תוצאה טובה עם אלגוריתם כל כך נאיבי.

הערות

1. אם נמלא את המשימות בסדר יורד לפי זמני ריצה $t_1\geq t_2\geq\ldots\geq t_n$ ונשתמש באלגוריתם נגיע ל-$\frac{3}{2}$ קירוב.
2. ניתן להשיג קירוב $1+\varepsilon$ בסמן ריצה $O\left(n^{\left(\frac{1}{\varepsilon}\right)^{\frac{3}{2}}}\right)$, למשל עבור $\varepsilon=\frac{1}{100}$ נקבל כי $n^{\left(\frac{1}{\varepsilon}\right)^{\frac{3}{2}}}=n^{1000}$.

13.1 העשרה - יישומים Multithreaded Algorithms

יתווסף בקרוב.

14 בעיית כיסוי על ידי קבוצות - Set Cover

14.0.0.1 סיפור מסגרת אנחנו נוסעים לקוטב הצפוני ויש לנו רשימה של דרישות. יש לנו מספר דמויות בעלות תכונות מסוימות. נרצה לשלוח כמה שפחות אנשים ככה שנענה על כל הדרישות.

קלט n מספר טבעי ו-r תת קבוצות $A_1,\dots,A_r$ של $[n]$ כך ש-$\bigcup_{i=1}^{r} A_i=[n]$.

פלט תת קבוצה $S\subseteq[r]$ כך ש-$\bigcup_{i\in S} A_i=[n]$ וכך ש-$|S|$ מינימלי.

זו בעיה NP קשה ולכן נחפש אלגוריתם מקרב.

דוגמה. עם $r=6,10$

$$A_1=\{1,2,3,4,5\}, A_2=\{6,7,8,9,10\}, A_3=\{1,2,3\}, A_4=\{6,7,8\}, A_5=\{1,2,3,6,7,8\}, A_6=\{3,4,5\}$$

מה הפתרון האופטימלי? מספיקות שתי קבוצות $S^\star=\{1,2\}$, שכן לא מספיקה קבוצה אחת. נשאלת השאלה, איזו קבוצה נוכל להכניס ראשונה? נוכל לבחור באסטרטגיה חמדנית, לכן נתחיל עם קבוצה בגודל מקסימלי עם 5. נשנה את העולם, היינו נוציא את כל האיברים המשותפים בין הקבוצות הנותרות ל-A_5. מכאן נבחר ב-$6,2$ ונקבל $S=\{5,6,2\}$. זה לא אופטימלי. אבל נבדוק כמה הוא מקרב. על כן, נרשום:

אלגוריתם 11 אלגוריתם מקרב לבעיית כיסוי הקבוצות

העקרון החמדני: בכל שלב נוסיף קבוצה המכסה הכי הרבה ממה שנשאר.

- אתחול: $G=\emptyset$ הפתרון החמדן, $X=[n]$ מה שנותר לכסות.
- איטרציה: יהי $1\le i^\star\le r$ כך ש-$|A_{i^\star}\cap X|=\max_{1\le i\le r}|A_i\cap X|$ נעדכן $G=G\cup\{i^\star\}$ ו-$X=X\backslash A_{i^\star}$.
- סיום: כאשר $X=\emptyset$ נעצור ונחזיר את G.

שאלה איזה קירוב האלגוריתם הזה משיג?

תשובה $\lceil\ln n\rceil$.

טענה. (העשרה) כל קירוב שהוא יותר טוב מ-$\Omega(\ln n)$ הוא כבר NP קשה להשגה.

הערה. בהמשך נראה דוגמאות כל כך קשות שעדיף פשוט לבחור באופן ראנדומי.

משפט. האלגוריתם החמדני משיג $\lceil\ln n\rceil$ קירוב לבעיה.

ניתן מעט אינטואיציה לפני ההוכחה. נסמן ב-X_j את מה שנשאר לכסות אחרי j איטרציות. נרשום את צעדי האלגוריתם מהדוגמא הקודמת:

$$|X_0|=10$$
$$|X_1|=4$$
$$|X_2|=2$$
$$|X_3|=0$$

נבחין כי בכל פעם, הקבוצה הנותרה להוספה קטנה לפחות בחצי מהפעם הקודמת. למה זה נכון? אנו יודעים שיש פתרון אופטימלי בגודל 2, על כן יש שתי תתי קבוצות שמכסות את כל X, לכן הגודל של אחת גדול או שווה מ-$\frac{|X|}{2}$ ולכן הקבוצה שאנו בחרנו בגודל

גדול יותר ולכן לאחר שנחסר אותה מ-X נקבל קבוצה חדשה בגודל $\frac{|X|}{2}$. נביט במה שנשאר מאותן שתי קבוצות. הן עדיין מכסות את כל העולם, ולכן עוד הפעם נוכל לבחור קבוצה שתקטין בחצי, ככה נמשיך עד שנסיים למלא. לכן יש $\log_2 |X|$ צעדים סך הכל. כאן הבסיס היה 2 כי הפתרון האופטימלי היה בגודל 2.

עתה נניח כי הפתרון האופטימלי בגודל k. לכן יש לנו $A_1, \ldots, A_k$ שמכסות את $[r]$. לכן יש אחת שמכסה $\frac{1}{k}$ מהעולם, לכן נשאר לנו $\left(1-\frac{1}{k}\right) n$, לאחר הצעד השני עוד הפעם נחסיר לפחות $\frac{1}{k}$ מהעולם ונקבל $\left(1-\frac{1}{k}\right)^2 n$, מה החזקה המינימלית כדי להגיע ל-1? יהי t חזקה זו אזי

$$\left(1-\frac{1}{k}\right)^t n < 1$$

זה קורה אם"ם

$$\left(1-\frac{1}{k}\right)^t < \frac{1}{n}$$

עבור $t = k \ln n$ מתקבל כי

$$\left(1-\frac{1}{k}\right)^{k \ln n} < e^{-\ln n} = \frac{1}{n}$$

כרצוי. לכן הפתרון הוא מסדר גודל של $k \ln n$. לכן היחס הוא מסדר גודל של $\ln n$.

לפני הוכחת המשפט ננסח שתי למות.

סימונים נסמן ב-$S^\star$ פתרון אופטימלי, $k = |S^\star|$, נסמן G הפתרון החמדן ו-$t = |G|$. עבור $0 \le j \le t$ נסמן ב-X_j את הקבוצה שנותר לכסות אחרי j האיטרציות הראשונות של האלגוריתם, אזי $[n] = X_0 \ge X_1 \ge \ldots \ge X_t = \emptyset$. נשים לב כי t הוא גם מספר האיטרציות של האלגוריתם עד העצירה.

דוגמה. עם $r = 7, n = 9$

$$A_1 = \{1,2,3\}, A_2 = \{4,5,6\}, A_3 = \{7,8,9\}, A_4 = \{1,2,4,5\}, A_5 = \{1,2\}, A_6 = \{4,5\}, A_7 = \{8,9\}$$

אזי

$$\begin{aligned} X_0 &= [9] \\ X_1 &= \{3,6,7,8,9\} \\ X_2 &= \{3,6\} \\ X_3 &= \emptyset \end{aligned}$$

עם $S^\star = \{1,2,3\}, S = \{4,3,1\}$.

למה. (1) לכל $y > 0$ ולכל $0 < x < 1$ מתקיים $(1-x)^y < e^{-xy}$.

הוכחה: (העשרה) קיים $r > 1$ כך ש-$x = \frac{1}{r}$ ולכן אי השוויון שקול ל-$\left(1-\frac{1}{r}\right)^y < e^{-\frac{y}{r}}$. נבחין כי $\left(1-\frac{1}{r}\right)^r < e^{-1}$ ולכן $\left(1-\frac{1}{r}\right)^{ry\frac{1}{r}} < e^{-y\frac{1}{r}}$ כרצוי.

למה. (2) לכל $0 \le y \le t-1$ מתקיים כי $\max_{i \in S^\star} |A_i \cap X_i| \ge \frac{1}{k}|X_j|$.

הוכחה: נראה כי $\bigcup_{i\in S^\star}(A_i\cap X_j)=X_i$. אמנם מכיוון ש-$S^\star$ הוא פתרון חוקי, מתקיים $\bigcup_{i\in S^\star}A_i=[n]$ ולכן

$$\bigcup_{i\in S^\star}(A_i\cap X_j)=\left(\bigcup_{i\in S^\star}A_i\right)\cap X_j=[n]\cap X_j=X_j$$

כרצוי. עתה,

$$\max_{i\in S^\star}|A_i\cap X_j|\overset{\text{ממוצע קטן ממקסימום}}{\geq}\frac{1}{k}\sum_{i\in S^\star}|A_i\cap X_j|\overset{\text{חסם האיחוד}}{\geq}\frac{1}{k}\left|\bigcup_{i\in S^\star}(A_i\cap X_j)\right|=\frac{1}{k}|X_j|$$

וסיימנו. ■ ■

הוכחה: (המשפט) נזכור כי $k=|S^\star|$ וכי $t=|G|$. צריך להראות כי $|G|\leq|S^\star|\cdot\lceil\ln n\rceil$. כלומר, נסמן $u=k\lceil\ln n\rceil$ ונוכיח כי $t\leq u$. נניח בשלילה. אזי $|X_n|\geq 1$ נשים לב שמלמה 2 נובע כי לכל $0\leq j\leq t-1$ מתקיים $|X_{j+1}|\leq\left(1-\frac{1}{k}\right)X_j$. נוכיח זאת. אמנם, לפי עקרון הפעולה של האלגוריתם החמדן, יהי $i^\star$ האינדקס שהאלגוריתם בחר באיטרציה ה-$j+1$. לכן

$$|A_{i^\star}\cap X_j|=\max_{1\leq i\leq r}|A_i\cap X_j|\geq\max_{i\in S^\star}|A_i\cap X_j|\overset{\text{למה 2}}{\geq}\frac{1}{k}|X_j|$$

ולכן

$$|X_{j+1}|=|X_j\backslash A_{i^\star}|=|X_j|-|X_j\cap A_{i^\star}|\leq|X_j|-\frac{1}{k}X_j=\left(1-\frac{1}{k}\right)X_j$$

נניח בשלילה כי $t>u$ ונגיע לסתירה. אם $t>u$ אזי $X_u\neq\emptyset$ ולכן

$$\begin{aligned}1\leq|X_u|&\leq\left(1-\frac{1}{k}\right)|X_{u-1}|\leq\ldots\leq\left(1-\frac{1}{k}\right)^u X_0\overset{\text{למה 1}}{<}e^{-\frac{u}{k}}|X_0|\\&=e^{-\frac{k\lceil\ln n\rceil}{k}}\cdot|X_0|\leq e^{-\ln n}|X_0|=\frac{1}{n}\cdot n=1\end{aligned}$$

סתירה. ■

15 תרגול 6 - תכנון ליניארי (LP)

15.1 מבוא

עד כה עסקנו בבעיות אופטימיזציה. הייתה לנו פונקציית מטרה ומרחב פתרונות תקין. בתכנון ליניארי ופונקציית המטרה היא ליניארית שתלויה באופן ליניארי במשתניה.

הגדרה. בעיית אופטימיזציה תקרא בעיית תכנון ליניארי אם ניתן לכתוב אותה בצורה הבאה:

$$\begin{aligned}&\max c^Tx\\s.t\ &Ax\leq b\\&x\geq 0\end{aligned}$$

כאשר אי השוויו מוגדר לכל קוארדינטה.

הערה. אם נרצה מינימום במקום מקסימום נכפול הכל ב--1.

דוגמה. הבעיה הבאה

$$\begin{aligned}
&\max x_1 + 2x_2\\
&4x_1 - x_2 \leq 3\\
&2x_1 + 2x_2 \geq -100\\
&x_1 \geq 0\\
&x_2 \geq 0
\end{aligned}$$

היא בעיית תכנות לינארי. שכן נוכל לרשום

$$\begin{aligned}
&\max \begin{pmatrix} 1 \\ 2 \end{pmatrix}^T \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\\
&\begin{pmatrix} 4 & -1 \\ -2 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 3 \\ -100 \end{pmatrix}\\
&\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \geq 0_{\mathbb{R}^2}
\end{aligned}$$

כרצוי.

יש אלגוריתמים לפתרון בעיות כאלה:

- Simplex Algorithm
- Ellipsoid Algorithm
- Karmarker's Algorithm.

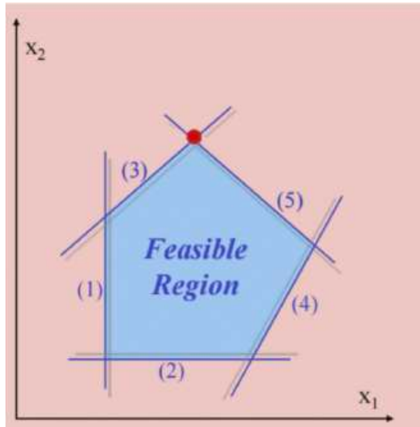
השלושה פותחו בזמן המלחמה הקרה בין ברית המועצות לארה"ב. זמן הריצה של הטוב ביותר הוא פולינומיאלי ב-$m+n$. השורה התחתונה היא שיש אלגוריתם יעיל ולכן נוכל להשתמש בו.

15.1.1 הגאומטריה של בעיית תכנון לינארי

דוגמה. נביט בבעיה

$$\begin{aligned}
&\max x_1 + 8x_2\\
s.t\ &x_1 \geq 3\\
&x_2 \geq 2\\
&-3x_1 + 4x_2 \leq 14\\
&4x_1 - 3x_2 \leq 25\\
&x_1 + x_2 \leq 15
\end{aligned}$$

אם במקום אי שוויון היה שוויון, מה היינו מקבלים בעצם? היינו מקבלים מישור, או פרטי פוליהדרון.



איור 4: התחום הכחול הוא תחום הפתרונות החוקיים. כדי למצוא מקסימום, נרצה ללכת בכיוון הגרדיאנט של הקלט לפונקציית הערך, דהיינו $\nabla\left(x_1+8x_2\right)$, כמה שיותר. דבר זה מוביל אותנו להבנה שמשתלם להגיע לגבולות התחום.

הגדרה. (Hyperplane, Halfspace) יהיו וקטור ממשי $a_j \in \mathbb{R}^n$ וסקלר b_j. הקבוצה $\left\{x \in \mathbb{R}^n \mid a_j^T x = b_j\right\}$ נקראת <u>על מישור</u>. הקבוצה $\left\{x \in \mathbb{R}^n \mid a_j^T x \leq b_j\right\}$ נקראת חצי מרחב.

הגדרה. (Polyhedron) קבוצת נקודות שניתן לתאר בצורה $\{x \in \mathbb{R}^n \mid Ax \leq b\}$ עבור $x \in \mathbb{R}^{m\times n}$ ו-$b \in \mathbb{R}^m$ נקראת <u>פוליהדרון</u>.

למעשה, אם נסמן את השורה ה-j של A על ידי a_j ואת האיבר ה-j ב-b על ידי b_j נראה שלמעשה כל חיתוך של m חצאי מרחבים מגדיר פוליהדרון, לכן קבוצות הפתרונות החוקיים של בעיית תכנון לינארי $\{x \geq 0 \mid Ax \leq b\}$ היא פוליהדרון הבנוי מחיתוך $m+n$ חצאי מרחבים. לכן תכנון לינארי הוא למעשה מיקסום של פונקציה לינארית על פני פוליהדרון. מסתבר שהפתרון של בעיות כאלה תמיד מתקבל בפינות של הפוליהדרון, וזהו הבסיס לאחר האלגוריתמים הפופליריים ביותר לפתרון בעיות תכנון לינארי, אלגוריתם ה-Simplex[2] שאותו לא נלמד.

15.1.2 פתרון בעיות תכנון לינארי

בעיות תכנות לינארי ניתנות לפתרון בזמן פולינומיאלי ב-$n+m$. ההוכחה לעובדה זו ניתנה ב-1979 על ידי Kachian שבהראה כי אלגוריתם האליפסואיד פותר את הבעיה בזמן פולינומיאלי כיום ידועים כמה אלגוריתם אשר פתורים בעיות תכנון לינארי בזמן פולינומיאלי. כשנראה את שיטת הקירוב על ידי תכנון לינארי, נשתמש בעובדה שניתן לפתור את הבעיה בזמן פולינומיאלי כדי לתת אלגוריתם קירוב יעיל לבעיה קשה.

המטרה שלנו היא לקרב אלגוריתמים באמצעות אלגוריתמי תכנון לינארי.

15.2 התרמיל השברי

קלט $\{(v_1,w_1),\ldots\}$ עם $w \in \mathbb{R}_+$.

פלט $x \in \mathbb{R}^n$ כך ש-$x = \operatorname{argmax} \sum_{i\in[n]} x_i v_i$ וגם $\sum_{i\in[n]} x_i w_i \leq W$ כאשר $0 \leq x_i \leq 1$ לכל $1 \leq i \leq n$.

הערה. מתקיים כי $\max_x c^T x = \max_x \langle c, x\rangle$.

איך נהפוך את הבעיה לתכנון לינארי? נרשום

$$\max \sum_{i=1}^{n} x_i v_i$$
$$\sum_{i=1}^{n} x_i w_i \leq W$$
$$0 \leq x_i \leq 1$$

[2] Dantzig, 47

נרשום את תנאי החוקיות $Ax \leq b$ על ידי

$$Ax = \begin{bmatrix} w_1 & \dots & \dots & \dots & w_n \\ 1 & 0 & \dots & \dots & 0 \\ 0 & \ddots & & \vdots & \\ \vdots & & & \ddots & \\ 0 & & & & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \leq \begin{pmatrix} W \\ \vdots \\ W \end{pmatrix} = b$$

וגם $c = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$ לתנאי המקסימום. על כן זו בעיית תכנון לינארי. אנו יודעים לפתור אותה, האם נוכל לדרוש כי $x \in \mathbb{Z}^n$? אנו יודעים שמבחינה גאומטרית יתכן כי הקודקוד לא יהיה מספר שלם, ולכן נצטרך לעגל את התוצאה ולקבל קירוב, כאשר העיגול יעשה בזהירות כדי שנקבל פתרון חוקי.

הגדרה. בעיית אופטימיזציה תקרא בעיית Integer Linear Programming (ILP), אם ניתן לכתוב אותו בצורה הבאה:

$$\begin{aligned} \max c^T x \\ s.t\ Ax \leq b \\ x \geq 0 \\ x \in \mathbb{Z}^n \end{aligned}$$

הערה. מסתבר שבעיות אופטימיזציה רבות הן בעיות ILP, אך לצערנו בעיות כאלה הן NP קשות. יחד עם זאת, נוכל לפתור בעיית LP ולעגל לפתרון ILP ובכך לקבל קירוב.

15.3 כלים לקירוב

נריץ את האלגוריתם הבא:

אלגוריתם 12 אלגוריתם לקירוב ILP באמצעות LP

1. אם הבעיה היא בעיית ILP, נעשה רקלסציה בכך שנוריד את הדרישה $x \in \mathbb{Z}^n$.
2. נריץ פותר LP ונקבל פתרון x_f. אם הפתרון בשלמים נחזיר אותו.
3. אחרת, נעגל את x_f לפתרון בשלמים חוקי ונחזיר את התוצאה.

15.4 בעיית הסרת המשולשים

הגדרה. יהי $G = (V, E)$ משולש הוא שלשה $u, v, w \in V$ כך ש-$\{u, v\}, \{u, w\}, \{v, w\} \in E$.

קלט גרף לא מכוון $G = (V, E)$.

פלט $S \subseteq E$ כך שב-$(V, E \backslash S)$ אין משולשים ו-S בגודל מינימלי.

נרצה לתאר זאת בצורה לינארית. ההגבלה שלנו היא שאין יותר משולשים, ונרצה להגדיר פונקציית משקל. נרצה לקבל

$$\begin{aligned} \min |S| \\ s.t\ \forall u, v, w \in V : s.t\ \{v, u\}, \{v, w\}, \{u, w\} \in E : \{u, v\} \in S \vee \{v, w\} \in S \vee \{u, w\} \in S \end{aligned}$$

אז נרצה להגדיר מעין וקטור אינדיקטור שבעזרתו נדע האם אנו זורקים את הצלע או לא.

נחשב על וקטור $x\in\{0,1\}^m$ כך ש-$x_e=1$ אם"ם $e\in S$ ונרצה לקבל $\min\sum\limits_{e\in E} x_e$, כלומר למזער סכום זה. נרצה לדרוש כי כל משולש מכיל לפחות צלע אחת שב-S, כלומר נסיר לפחות צלע אחת מתוך המשולש. נקבל אם כך:

$$\forall u,v,w\in V:\{v,u\},\{v,w\},\{u,w\}\in E: x_{\{u,v\}}+x_{\{v,w\}}+x_{\{u,w\}}\geq 1$$
$$\forall e\ 0\leq x_e\leq 1$$

כאן דרשנו $x_e\in[0,1]$ למרות שרצינו פתרון ב-$\mathbb{Z}^n$, אבל מכיוון שמדובר בבעיה NP קשה, נקרב לפי האלגוריתם. נרצה להמיר את מה שמצאנו לבעיית תכנון לינארי. נגדיר $c=\begin{pmatrix}-1\\ \vdots\\ -1\end{pmatrix}$ שכן אנו רוצים מינימיזציה. עתה, המטריצה A צריך לכלול משולשים כאשר כל שורה תייצג משולש. על כן $A=\left[\begin{array}{ccccc}0 & \dots & -1 & \dots & 0\\ & & & & \\ \hline 1 & & & & 0\\ & & 1 & & \\ 0 & & & & 1\end{array}\right]$ כאשר הבלוק הראשון מייצג בכל שורה משולש, ומכיל -1 באינדקס המתאים לצלע המתאימה במשולש, גודלו מסדר גודל של $|V|^3$. הבלוק השני מכיל מטריצת יחידה כדי שנוכל לרשום תנאי על הגודל של x_i. על כן, נקבל כי $\left(\text{argmax}\sum -x_e=\text{argmin}\sum x_e\right)$ $\text{argmax}_{x\in\mathbb{R}^m}\sum_{e\in E}-x_e=-\text{argmin}\sum_{e\in E}x_e$. לכל משולש u,v,w נקבל כי $-x_{\{u,v\}}-x_{\{u,w\}}-x_{\{v,w\}}\leq -1$ במטריצה, ו-$1\geq x_e$. דהיינו עבור $b=\begin{pmatrix}-1\\ \vdots\\ -1\\ 1\\ \vdots\\ 1\end{pmatrix}$ נקבל את הבעיה

$$\max c^T x$$
$$Ax\leq b$$
$$x\geq 0$$

כרצוי.

15.5 אלגוריתמי קירוב

הגדרה. (בעיית אופטימיזציה) נרצה אלגוריתם שבהנתן קלט x מחזיר פתרון אופטימלי, מקסימלי, מינימלי ביחס לפונקציית מטרה, מבין הפתרונות החוקיים. לכל x נגדיר X כסט הפתרונות החוקיים שלו.

הגדרה. בהנתן X ופונקציית מטרה f, נרצה למצוא $\max\limits_{x^\star\in X} f(x^\star)$. פתרון הוא <u>מקרב $c\geq 1$</u> אם לכל X הוא מחזיר $x\in X$ כך ש-$f(x)\geq\frac{1}{c}f(x^\star)$. עבור מינימיזציה, $f(x)\leq cf(x^\star)$ עבור $x^\star=\text{argmin}_{x\in X}f(x)$.

לרוב נקצר ונסמן את ערך הפתרון האופטימלי עבור בעיה נתונה ב-opt. נבחין כי באלגוריתמי קירוב לבעיות מינימיזציה (מקסימיזציה), הפתרון של האלגוריתם שלנו תמיד יהיה גדול או שווה (קטן או שווה) מהפתרון של האלגוריתם האופטימלי, מהגדרת האופטימליות.

16 בעיית כיסוי על ידי קודקודים Vertex Cover

קלט גרף $G=(V,E)$.

פלט תת קבוצה $S\subseteq V$ כך שלכל צלע $e=(x,y)\in E$ מתקיים $x\in S$ או $y\in S$ וכך ש-$|S|$ מינימלי.

זו בעיה NP קשה. נחפש אלגוריתם קירוב.

דוגמאות

1. $|S^\star|=1$ עם גרף כוכב, שמקודקוד אחד יוצאות צלעות לכל שאר הקודקודים.
2. $G=K_n$ הגרף השלם, אזי $|S^\star|=n-1$.
3. $G=K_{a,b}$ עם $a\le b$ כלומר גרף שלם דו צדדי. אנו יודעים כי $|S^\star|\le a$ כי אין צלעות בין הצדדים. יתר על כן, בהכרח $|S^\star|=a$ אחרת בהכרח בחרנו קודקוד מימין, כי אם לא, נקבל כי יש צלע עם קודקוד מ-a שלא כיסינו. אבל מכך שבחרנו פחות מ-a צלעות נוכל להראות שהקודקוד שלא בחרנו מ-V_a מכיל צלע לקודקוד ב-b שלא בחרנו.

נרצה לרשום אלגוריתם כללי לפתרון הבעיה.

אבחנה ניתן "לעשות רדוקציה" מהבעיה הזאת לבעיית כיסוי על ידי קבוצות.

כל $x\in V$ מכסה את קבוצת הצלעות העוברות דרכו. נסמן ב-A_x את קבוצת הצלעות הזאת. נתונות לנו תת קבוצות של צלעות $\{A_x\}_{x\in V}$ ורוצים לכסות את כל הצלעות E על ידי כמה שפחות מהקבוצות האלה. האלגוריתם ל set cover יגיד: בכל שלב נבחר קודקוד שמכסה הכי הרבה מהצלעות שלא כיסינו וזה ישיג $\lceil\ln(n)\rceil$ קירוב.

שאלה מכיוון שמדובר במקרה פרטי, האם האלגוריתם יתן תוצאה טובה יותר?

תשובה אפשר למצוא דוגמא לגרף שיתן קירוב $\ln n\approx\frac{1}{n}\ldots+\frac{1}{2}+1$.

האלגוריתם שלנו יהיה יותר מתוחכם.

אלגוריתם 13 אלגוריתם לא חמדני

- <u>אתחול</u> $S=\emptyset$ ו-$X=E$.
- <u>איטרציה</u> נבחר שרירותית צלע $X\ni e=(x,y)$ ונעדכן $S=S\cup\{x\}\cup\{y\}$. נמחק מ-X את כל הצלעות שעוברות דרך x או דרך y.
- <u>סיום</u>: כאשר $X=\emptyset$ נעצור ונחזיר את S.

דוגמה. (כשלון האלגוריתם) עבור הגרף $1\to3\to2\to4$ גרף לא מכוון. נקבל כי האלגוריתם יחזיר את $S=\{1,4,2,3\}$. למרות ש-$S^\star=\{1,2\}$.

האלגוריתם הנ"ל מייצג מגוון גדול של בעיות במדעי המחשב. ההשערה היא שכל קירוב טוב יותר מ-$\sqrt{2}$ הוא NP קשה. אנו נוכיח שהאלגוריתם משיג 2 קירוב.

משפט. האלגוריתם משיג 2-קירוב לבעיה, כלומר אם $S^\star$ הוא פתרון אופטימלי ואם S הוא הפתרון המוחזר על ידי האלגוריתם אזי $|S|\le2|S^\star|$.

הגדרה. <u>זיווג בגרף</u> הוא אוסף צלעות בגרף ללא קודקודים משותפים.

למה. אם הגרף G מכיל זיווג עם t צלעות, אזי כל כיסוי על ידי קודקודים ב-G מכיל לפחות t קודקודים.

הוכחה: יהי S כיסוי קודקודים ב-G. אזי S מכיל קודקוד על כל אחת מ-t צלעות הזיווג. מכיוון שלצלעות הזיווג אין קודקודים משותפים, כל t הקודקודים האלה שונים זה מזה ולכן S מכיל לפחות t קודקודים. ■

הוכחה: (המשפט) נסמן ב-$S^\star$ אופטימלי. נסמן ב-t את מספר האיטריצות של האלגוריתם עד העצירה. עבור $1\le j\le t$ נסמן ב-t_j את הצלע שהאלגוריתם בוחר באיטרציה ה-j. נסמן ב-S את הפתרון שהאלגוריתם יחזיר. נשים לב ש-$|S|=2t$, שכן כל פעם הוספנו קודקודים שלא מופיעים באף צלע קודמת שעברנו עליה. נשים לב גם כי על פי דרך פעולתו של האלגוריתם מתקיים שלצלעות $e_1,e_2,\ldots,e_t$ אין קודקודים משתותפים ולכן הן מהוות זיווג בגרף עם t צלעות ולכן לפי הלמה שהוכחנו מתקיים $|S^\star|\ge t$ ולכן $t\le|S^\star|\le|S|=2t\le2|S^\star|$. ■

17 בעיית כיסוי קודקודים ממושקל

קלט גרף פשוט $G = (V, E)$ ופונקציית משקל $w : V \to \mathbb{R}^+$

פלט כיסוי קודקודים $S \subseteq V$ עם משקל מינימלי כאשר $w(S) = \sum_{v \in S} w(v)$.

זו בעיה NP קשה, לכן נרצה לתת אלגוריתם מקרב לבעיה. נבחין כי זו הכללה של הבעיה הקודמת עם $w(v) = 1$. איזו אלגוריתם נציע לפתרון הבעיה?

הצעה. נבצע בדיוק אותו הדבר כמו בבעיה הקודמת, אבל נבחר את הצלע הכי זולה.

זה לא יעבוד. עבור הגרף $\begin{matrix} 2 & 3 \\ 1 & 4 \end{matrix}$ עם המשקלים $w(2) = w(1) = 1, w(3) = w(4) = 10^6$ נקבל שהמשקל גדול מ-10^6 למרות שיש פתרון עם ערך 2.

נוכל למשקל את הקודקודים לפי המשקל חלקי כמות הצלעות ולבחור בכל פעם ערך מינימלי. יתכן שיעבוד, אבל לא בהכרח.

17.1 פתרון באמצעות תכנון לינארי

נוכל לפתור בעיה זו בעזרת תכנון לינארי, ולקבל 2 קירוב. זאת נבצע על ידי ניסוח LP לבעיה ולהשתמש ב-$LP\ Solver$. רוב פותרים אלה משתמשים ב-$Simplex$. אנחנו מסתכלים על פאון במישור ועל קודקודיו, אפשר להוכיח שהפתרון האופטימלי מופיע על הקודקודים. אלגוריתם זה בוחר קודקוד מסויים, בצורה יעילה ומשווה לקודקודים שמצדדיו, אם הוא גדול מהם, הוא המקסימום מבין כולם - אפשר להוכיח. אחרת הוא עובר לאחד שהוא קטן ממנו וממשיך הלאה. זוהי רק הוריסטיקה, לא הוכיחו נכונות.

בסוף שנות ה-70 מדען רוסי הציע אלגוריתם שניתן להוכיח נכונותו. הנשיא קרטר כל כך התלהב שהוא התקשר אליו באופן אישי.

היתרון של התכנון הלינארי הוא שהוא מאפשר לקרב בעיות רבות בצורה מאוד מכנית. נבצע זאת על פי האסטרטגיה הבאה:

אלגוריתם 14 פתרון בעיות באמצעות תכנון לינארי

1. נתרגם את הבעיה לבעיה שקולה של בעיית אופטימיזציה (מינימיזציה) של פונקציה פונקציה לינארית על אוסף של משתנים המקיימים אילוצים לינאריים והמקבלים ערכים שלמים - ILP. דהיינו נעביר בעיה NP קשה לבעיה NP קשה אחרת.
2. נחליף את האילוץ של ערכים בשלמים באי שוויונים לינארים רגילים. **חשוב:** האילוצים הלינארים החדשים צריכים להיות מחמירים פחות מהאילוץ בשלמים. כלומר הבעיה החדשה צריכה להיות מתירנית יותר. לדוגמא, במקום $X \in \{0, 1\}$ נדרוש $0 \leq X \leq 1$. לבעיה זו נקרא רקלסציית-LP של הבעיה המקורית.
3. נפתור את בעיית ה-LP שקיבלנו ונקבל פתרון אופטימלי שלה, באמצעות אלגוריתם Black Box כלשהו.
4. <u>עיגול</u>: נעגל את הפתרון שקיבלנו בשלב 3 לפתרון טוב של הבעיה המקורית.

כיצד נבצע זאת על הדוגמא שלנו? אנחנו רוצים למצוא כיסוי, כדי לבצע זאת, לכל קודקוד נתאים משתנה מתאים שיציין האם הוא בכיסוי או לא. נרשום בצורה לינארית את גודל הכיסוי ונרצה למצוא מינימיזציה לכמות הקודקודים בכיסוי.

אלגוריתם 15 אלגוריתם 2 מקרב לפתרון הבעיה

1. בניית תכנית ILP : לכל $v\in V$ מתאים משתנה $X\left(v\right)$ בהינתן כיסוי קודקודים S נגדיר $X\left(v\right)=\begin{cases}1 & v\in S\\ 0 & v\notin S\end{cases}$. בהינתן צלע $\left(a,b\right)=e\in E$ מתקיים ש-$a\in S\vee b\in S$ או $X\left(a\right)=1\vee X\left(b\right)=1$. ניתן לכתוב זאת כאילוץ לינארי אחד על המשתנים שלו $X\left(a\right)+X\left(b\right)\geq1$. מתקיים גם כי משקל הכיסוי Sהוא פונקציה לינארית על המשתנים שהגדרנו כלומר $w\left(S\right)=\sum\limits_{v\in S}w\left(v\right)=\sum\limits_{v\in V}w\left(v\right)X\left(v\right)=X^{T}\cdot w$. קיבלנו את תכנית ה-ILP הבאה:

$$\text{ILP}\begin{cases}\min\sum\limits_{v\in V}w\left(v\right)X\left(v\right)\\ \forall v\in V:X\left(v\right)\in\left\{0,1\right\}\\ \forall\left(a,b\right)=e\in E:X\left(a\right)+X\left(b\right)\geq1\end{cases}$$

2. רקלסציה: נחליף לכל $v\in V$ את האילוץ $X\left(v\right)\in\left\{0,1\right\}$ לאילוץ לינארי $0\leq X\left(v\right)\leq1$. קיבלנו את התכנית הלינארית הבאה. קיבלנו את תכנית ה-LP הבאה:

$$\text{ILP}\begin{cases}\min\sum\limits_{v\in V}w\left(v\right)X\left(v\right)\\ \forall v\in V:X\left(v\right)\in\left[0,1\right]\\ \forall\left(a,b\right)=e\in E:X\left(a\right)+X\left(b\right)\geq1\end{cases}$$

3. פתרון LP: נפתור את בעיית ה-LP שקילבנו בשלב 2 ונקבל פתרון אופטימלי שלה. נסמן ב-$X^{\star}=\left\{X^{\star}\left(v\right)\right\}_{v\in V}$ פתרון אופטימלי של הבעיה המקורית, ונסמן $X_{LP}^{\star}$ את הפתרון האופטימלי שקיבלנו לבעיית ה-LP בשלב זה.

4. עיגול: לכל $v\in V$ נגדיר $X\left(v\right)=\begin{cases}1 & X_{LP}^{\star}\left(v\right)\geq\frac{1}{2}\\ 0 & X_{LP}^{\star}\left(v\right)<\frac{1}{2}\end{cases}$. נחזיר את X.

נותר להוכיח את נכונות האלגוריתם.

משפט. הפתרון X שהאלגוריתם מחזיר הוא פתרון חוקי ו-2 מקרב לבעיה המקורית. כלומר $w\left(X\right)=\sum\limits_{v\in V}w\left(v\right)\cdot X\left(v\right)\leq2\cdot\sum\limits_{v\in V}w\left(v\right)X^{\star}\left(v\right)=w\left(X^{\star}\right)$.

הוכחה: נוכיח חוקיות וקירוב.

חוקיות: על פי הגדרת X, לכל $v\in V$ מתקיים $X\left(v\right)\in\left\{0,1\right\}$. תהי $\left(a,b\right)=e\in E$. מכיוון ש-$X_{LP}^{\star}$ פתרון חוקי של בעיית ה-LP ולכן $X_{LP}^{\star}\left(a\right)+X_{LP}^{\star}\left(b\right)\geq1$ לכן $X_{LP}^{\star}\left(a\right)\geq\frac{1}{2}$ או $X_{LP}^{\star}\left(b\right)\geq\frac{1}{2}$ ולכן $X\left(a\right)=1$ או $X\left(b\right)=1$ ולכן $X\left(a\right)+X\left(b\right)\geq1$.

קירוב: נטען שתי למות.

למה. (1, חסם על האופטימלי) מתקיים $\sum\limits_{v\in V}w\left(v\right)\cdot X_{LP}^{\star}\left(v\right)\leq\sum\limits_{v\in V}w\left(v\right)X^{\star}\left(v\right)$

הוכחה: (למה 1) $X^{\star}$ הוא פתרון חוקי של בעיית ה-ILP, כל פתרון של בעיית ILP הוא גם פתרון חוקי של בעיית ה-LP ולכן $X^{\star}$ פתרון חוקי לבעיית ה-LP ואילו $X_{LP}^{\star}$ הוא פתרון אופטימלי לבעיית ה-LP. ■

למה. (2, חסם על העיגול) לכל $v\in V$ מתקיים $X\left(v\right)\leq2\cdot X_{LP}^{\star}\left(v\right)$. ההוכחה ברורה.

על כן,

$$\sum_{v\in V} w(v)\, X(v) \overset{\text{Lemma.2}}{\leq} \sum_{v\in V} w(v)\cdot 2\cdot X^\star_{LP}(v) = 2\sum_{v\in V} w(v)\, X^\star_{LP}(v)$$
$$\overset{\text{Lemma.1}}{\leq} 2\cdot \sum_{v\in V} w(v)\cdot X^\star(v)$$

כרצוי. ■

דוגמה. נביט בגרף מלא על שלושה קודקודים v_1, v_2, v_3 עם משקל 1 לכל קודקוד. ניתן ערך לכל קודקוד בין 0 ל-1 כך שסכום של כל צלע גדול מ-1, אבל הסכום הכולל מינימלי. ניתן ערך $\frac{1}{2}$ לכל אחד, אזי $\sum_{v\in V} w(v)\, X^\star_{LP}(v) = \frac{3}{2}$. נקבל כי $X^\star = (1,1,0)$ מינימלי עם $\sum_{v\in V} w(v)\, X^\star(v) = 2$.

18 תרגול 7 - אלגוריתמי קירוב, קירוב הסתברותי

18.1 אלגוריתמי קירוב

קיימות בעיות אופטימיזציה קשות רבות, שנוכל לתת להן פתרון מקרב יעיל במקום פתרון לא יעיל אופטימלי.

תהא בעיית מקסימיזציה המוגדרת באופן הבא. נסמן ב-X את מרחב הפתרונות החוקיים לבעיה וב-$f: X \to \mathbb{R}^+$ את פונקציית הערך של הבעיה. המטרה שלנו היא למצוא $x \in X$ הממקסם את $f(x)$, כלומר את $x^\star = \text{argmax}_{x\in X}\{f(x)\}$.

הגדרה. עבור כל $c \geq 1$ נאמר כי אלגוריתם הוא <u>אלגוריתם c-מקרב</u> לבעיית המקסימיזציה אם לכל קלט הוא פולט $x \in X$ כך שמתקיים $f(x) \geq \frac{1}{c} f(x^\star)$. אם הבעייה היא בעיית מינימיזציה אז $x^\star = \text{argmin}_{x\in X}\{f(x)\}$ ואלגוריתם c-מקרב ימצא $x \in X$ כך שמתקיים $f(x) \leq cf(x^\star)$. כמובן נדרוש בנוסף כי זמן ריצת האלגוריתם פולינומיאלי בגודל הקלט. לרוב נקצר ונסמן את ערך הפתרון האופטימלי עבור בעיה נתונה ב-$\text{OPT} = f(x^\star)$.

18.2 בעיית ה-3SAT (Statifiability)

קלט נוסחת $3-CNF$. עם m פסוקיות $C_1, \ldots, C_m$ מעל n משתנים $x_1, \ldots, x_n$. כאשר נוסחת CNF היא נוסחא מהצורה $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_4 \neg \vee x_5)$ בכל פסוקית יש 3 ליטרלים, בתוך הפסוקיות יש "או" בין הליטרילים, בין הפסוקיות יש "וגם".

פלט אם קיימת השמה ל-$x_1, \ldots, x_n$ כ שהנוסחא בעלת ערך T.

דוגמה. $x_1 = F, x_2 = F, x_3 = F, x_4 = F, x_5 = T$ אזי $\left(\underbrace{x_1 \vee x_2 \vee x_3}_{F\vee F\vee F}\right) \wedge \left(\underbrace{\neg x_1 \vee x_4 \neg \vee x_5}_{T\vee F\vee T}\right) = F \wedge T = F$.

הערה. זאת בעיית הכרעה (קיים, לא קיים) ולא בעיית אופטימיזציה.

הערה. זו בעייה NP-קשה שלא ידוע על פתרון יעיל לפתרונה.

18.2.1 בעיית Max-SAT

מכיוון שהבעיה הקודמת היא NP קשה, נרצה לקרב אותה, יחד עם זאת, זו לא בעיית אופטימיזציה, כדי לתקן זאת, נגדיר בעיה חדשה.

קלט בדיוק כמו ב-$3SAT$.

פלט השמה המספקת מספר מקסימלי של פסוקיות שערכן $\mathbb{T}$.

מסקנה. אם נפתור בעיה זו נוכל לפתור בוודאות את הבעיה הקודמת, שכן קיום פתרון מקסימלי בגודל מספר הפסוקיות מצביע על קיום פתרון, אחרת אי קיום.

לצערנו, גם זו בעיה NP קשה, אך אותה אפשר לקרב.

נציע אלגוריתם 2 מקרב לבעיה.

אלגוריתם 16 אלגוריתם 2-מקרב נאיבי לבעיה

---

1. נבדוק כמה פסוקיות ההשמה $\vec{X}_T = (T, \ldots, T)$ מספקת - נסמן t.
2. נבדוק כמה פסוקיות ההשמה $\vec{X}_F = (F, \ldots, F)$ מספקת - נסמן f.
3. אם $f < t$ נחזיר $\vec{X}_T$ אחרת נחזיר את $\vec{X}_F$

משפט. האלגוריתם הוא 2 מקרב לבעיה.

הוכחה: עלינו להוכיח חוקיות וקירוב.

חוקיות: האלגוריתם מחזיר את ההשמה $\vec{X}_T$ או $\vec{X}_F$ אשר הן השמות חוקיות.

קירוב: נראה כי $\max\{f,t\} \geq \frac{1}{2}m$. כל פסוקית מסופקת על ידי $\vec{X}_T$ או $\vec{X}_F$ או שניהם ולכן שתי הבחירות מחלקות את הפסוקיות לשתיים - מי שמסתפק עם F ומי שמסתפק עם T ולכן המקסימום מביניהם מספק לפחות חצי על כן $\max\{f,t\} \geq \frac{1}{2}m$, במילים אחרות, $f+t \geq m$ כי בהכרח סיפקנו את כל הפסוקיות, לכן $\max\{f,t\} \geq \frac{f+t}{2} \geq \frac{m}{2}$. כיוון ש-$\text{opt} \leq m$ נקבל כי $\max\{f,t\} \geq \frac{1}{2}m \geq \frac{1}{2}\text{opt}$. ■

הערה. נבחין כי האלגוריתם יעבוד עם כל השמה ושלילתה, הבחירה ב-$T,\ldots,T$ שרירותית.

הערה. האלגוריתם הוא 2 מקרב לכל נוסחת CNF, לא רק 3CNF.

השערה. יש אלגוריתם טוב יותר לפתרון הבעיה עבור 3CNF.

18.3 אלגוריתמי קירוב הסתברותיים

נראה כמה הגדרות לאלגוריתמים, כאשר כל אחד מחזיר פתרון חוקי בזמן פולינומיאלי.

הגדרה. אלגוריתם <u>דטרמיניסטי</u> מחזיר פתרון אופטימלי

הגדרה. אלגוריתם <u>c-מקרב</u> - מחזיר פתרון c-מקרב.

הגדרה. אלגוריתם <u>דטרמיניסטי הסתברותי</u> - מחזיר פתרון אופטימלי בהסתברות גבוהה כרצוננו אחרת מחזיר $fail$.

הגדרה. אלגוריתם <u>c-מקרב הסתברותי</u> - מחזיר פתרון c-מקרב בהסתברות גבוהה כרצוננו, אחרת מחזיר $fail$.

הגדרה. נאמר כי מאורע מתקיים ב-"הסתברות גבוהה כרצוננו" כאשר ניתן לרשום את הסתברותו כ-$1-\frac{1}{a^k}$ עבור $a>1, k\in\mathbb{N}$.

18.3.1 אלגוריתם $\frac{8}{7}$ מקרב ל-Max-3SAT

נציג את האלגוריתם הבסיסי:

אלגוריתם 17 אלגוריתם בסיסי

1. לכל משתנה x_i נטיל מטבע הוגן:
   (א) אם יצא עץ, נגדיר $x_i = \mathbb{T}$.
   (ב) אם יצא פלי, נגדיר $x_i = \mathbb{F}$.
2. אם ההשמה שהגרלנו מספקת לפחות $\frac{7}{8}m$ פסוקיות, נחזיר אותה, אחרת נחזיר $fail$.
3. <u>האלגוריתם הכללי:</u> נחזור על אלגוריתם הבסיסי $k\left(m+1\right)$ פעמים באופן בלתי תלוי. אם באחת הריצות הייתה הצלחה, נחזיר את ההשמה שהתקבלה, אחרת נחזיר $fail$.

שאלה מאיפה מגיע $\frac{7}{8}$?

תשובה כדי לקבל $\mathbb{T}$ בפסוקית, מספיק שאחד יהיה T. כדי שפסוקית תהיה $\mathbb{F}$ צריך שכולם יהיו F כלומר בהסתברות $\frac{1}{8}$.

משפט. האלגוריתם הוא $\frac{7}{8}$ מקרב לבעיה במקרה של הצלחה.

למה. הסיכוי שהאלגוריתם הבסיסי מצליח $\geq \frac{1}{m+1}$.

הוכחה: צריך להראות שהאלגוריתם הבסיסי מצליח בהסתברות מספיק גבוהה, ושבמידה והוא מצליח אז הוא $\frac{8}{7}$ מקרב.

<u>חוקיות:</u> במקרה של הצלחה, הפתרון כמובן חוקי, אחרת מוחזר $fail$.

<u>קירוב:</u> אם האלגוריתם מחזיר השמה ל-$x_1,\ldots,x_n$ שמספקת $\frac{7}{8}m$ פסוקיות אז הוא בבירור $\frac{8}{7}$ מקרב, שכן $m \geq \text{opt}$ ולכן $\frac{7}{8}m \geq \frac{7}{8}\text{opt}$.
נותר להוכיח כי הסתברות ההצלחה היא גבוהה מספיק, כלומר $1-\frac{1}{e^k}$. נסתמך על הלמה הבאה:
על כן $\mathbb{P}\left(\text{Success}\right) \geq \frac{1}{m+1}$. מכאן נסיק כי

$$\begin{aligned}\mathbb{P}\left(\text{האלגוריתם הכללי נכשל}\right) &= \mathbb{P}\left(\text{הלאוגריתם הבסיסי נכשל } k\left(m+1\right) \text{ פעמים}\right)\\ &= \mathbb{P}\left(\text{האלגוריתם הבסיסי נשכל}\right)^{k(m+1)}\\ &\leq \left(1-\frac{1}{m+1}\right)^{(m+1)k} = \left(\left(1-\frac{1}{m+1}\right)^{m+1}\right)^k\\ &< \left(\frac{1}{e}\right)^k = \frac{1}{e^k}\end{aligned}$$

מכאן נסיק כי האלגוריתם הכללי מצליח בסיכוי $> 1-\frac{1}{e^k}$. אי השוויון האחרון נובע מכך ש-$\frac{1}{e} = \lim_{n\to\infty}\left(1-\frac{1}{n}\right)^n$ שאיפה מלמטה.
■

הוכחה: (הלמה) שלבי ההוכחה יהיו כדלקמן:

1. נגדיר מ"מ Y שמייצגים את מספר הפסוקיות שלא סופקו ונחשב את $\mathbb{E}\left[Y\right]$.
2. נעזר באי שוויון מרקוב כדי להראות כי $\mathbb{P}\left(\text{האלגוריתם הבסיסי נכשל}\right) \leq \frac{m}{m+1}$.
3. נסיק כי $\mathbb{P}\left(\text{הלאוגירתם הבסיסי הצליח}\right) \geq \frac{1}{m+1}$.

נגדיר $Y:\Omega\to\mathbb{N}$ כאשר $\Omega = \left\{T,F\right\}^n$ עם $p\left(\omega\right) = \frac{1}{2^n}$ עם $\omega\in\Omega$, כך ש-מספר הפסוקיות שלא סופקו על ידי ההשמה ω $Y\left(\omega\right) = $.
נגדיר אם כך לכל פסוקית C_i נגדיר מ"מ מקרי

$$Y_i\left(\omega\right) = \begin{cases}1 & \omega\left(C_i\right) = \mathbb{T} \text{ סופק על ידי ההשמה}\\ 0 & \text{אחרת}\end{cases}$$

לכל $1 \leq i \leq m$. נשים לב כי $Y = \sum_{i=1}^{m} Y_i$. נזכיר כי $\mathbb{E}[X]$ התוחלת של X היא פעולה לינארית שמהווה ממוצע משוקלל על X. עתה, לכל פסוקית C_i מתקיים

$$\begin{aligned}\mathbb{E}[Y_i] &= 0 \cdot \mathbb{P}(Y_i = 0) + 1 \cdot \mathbb{P}(Y_i = 1) = \mathbb{P}(Y_i = 1) \\ &= \frac{1}{2^3} = \frac{1}{8}\end{aligned}$$

כאשר המעבר האחרון נובע מכך שעל מנת שהפסוקית ה-i לא תסופק צריך שכולם יהיו False. מכאן

$$\mathbb{E}[Y] = \mathbb{E}\left[\sum_{i=1}^{m} Y_i\right] = \sum_{i=1}^{m} \mathbb{E}[Y_i] = \sum_{i=1}^{m} \frac{1}{8} = \frac{m}{8}$$

עתה, נקבל כי

$$\begin{aligned}\mathbb{P}(\text{האלגוריתם הבסיסי נכשל}) &= \mathbb{P}\left(\frac{7}{8}m > \text{ההשמה סיפקה מספר פסוקיות}\right) \\ &= \mathbb{P}\left(\frac{1}{8}m > \text{ההשמה לא סיפקה מספר פסוקיות}\right) \\ &= \mathbb{P}\left(Y > \frac{1}{8}m\right) = \mathbb{P}(Y > \mathbb{E}[Y])\end{aligned}$$

ניזכר כי מאי שוויון מרקוב, עבור מ"מ אי שלילי $Y \geq 0$ ו-$c - 1 > 1$ נובע כי $\mathbb{P}(Y \geq c \cdot \mathbb{E}[Y]) \leq \frac{1}{c}$. כדי שנוכל להשתמש באי השוויון בצורה לא טריוויאלית, שכן ברור כי $\mathbb{P}\left(Y > \frac{1}{8}m\right) \leq 1$, נבחין כי $Y > \frac{1}{8}m \iff 8Y > m \iff 8Y \geq m + 1 \iff$ לכן $Y \geq \frac{1}{8}m + \frac{1}{8}$

$$\mathbb{P}\left(Y > \frac{1}{8}m\right) = \mathbb{P}\left(Y \geq \frac{1}{8}m + \frac{1}{8}\right) = \mathbb{P}\left(Y \geq \frac{1}{8}m \underbrace{\left(1 + \frac{1}{m}\right)}_{c}\right) \leq \frac{1}{1 + \frac{1}{m}} = \frac{m}{m+1}$$

ולכן ההסתברות להצלחה $\mathbb{P}(\text{Success}) = 1 - \mathbb{P}\left(Y > \frac{1}{8}m\right) \geq 1 - \frac{m}{m+1} = 1 - \left(1 - \frac{1}{m+1}\right) = \frac{1}{m+1}$.

באופן דומה, היינו יכולים להגדיר מ"מ שמציין פסוקיות שכן סופקו. ■

18.4 אלגוריתם ה-Simplex

19 בעיית פתרון אופטימלי של מערכת משוואות לינאריות מודולו 2 (MAX-LIM-2)

בעיה זו דומה לבעיית ה-MAX-3-SAT, לא זהה.

קלט מטריצה A עם k שורות ו-n עמודות ווקטור b באורך k מעל $\mathbb{F}_2$ המייצגת מערכת משוואות לינאריות $Ax = b$ כאשר $x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ הוא ווקטור משתנים.

פלט השמה $s = \begin{pmatrix} s_1 \\ \vdots \\ s_n \end{pmatrix}$ של ערכים למשתנים המספקת כמה שיותר משוואות.

דוגמה. $k = 3 = n$ והמערכת $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ כלומר המערכת $\begin{cases} x_1 = 1 \\ x_2 + x_3 = 1 \\ x_1 + x_2 + x_3 = 1 \end{cases}$. למערכת זו אין פתרון, כי אין הצגה המספקת את שתי הראשונות וגם את השלישית אחרת $x_1 + x_2 + x_3 = 2 \equiv 0 \neq 1 \mod 2$. לכן הפתרון האופטימלי הוא

$s = \left(\begin{smallmatrix}1\\1\\0\end{smallmatrix}\right)$.

זו בעיה NP קשה, נחפש אלגוריתם קירוב. ניקח השראה מבעיית MAX-3-SAT, ראינו כי השמה מקרית מספקת פסוקית אחת בסיכוי $\frac{7}{8}$ וכך קיבלנו אלגוריתם הסתברותי $\frac{8}{7}$ מקרב המחזיר $\frac{8}{7}$ קירוב בסיכוי גבוה כרצוננו.

נסתכל על משוואה אחת $x_1 + x_3 + x_{100} + x_{212} + x_{352} = 1$ מה הסיכוי שפתרון יקיים את זה? נוכל למלא בכל דרך אפשרית את הארבעה הראשונים ולבסוף לבחור את x_{352} כך שנקבל סכום 1, על כן חצי מהאפשרויות נותנות אחד ולכן נקבל זאת בסיכוי $\frac{1}{2}$.

ניתן אם כך אלגוריתם דטרמיניסטי שהוא 2 מקרב. האלגוריתם ישתמש בשיקולים הסתברותיים.

הערה. ההפיכה של אלגוריתם הסתברותי לדטרמיניסטי הוא נושא שלם במדעי המחשב. יש המאמינים שאפשר להפוך כל אלגוריתם הסתברותי לאלגוריתם דטרמיניסטי.

הגדרות וסימונים

נסמן ב-$\mathcal{S}$ את מרחב הפתרונות החוקיים לבעיה. כלומר $\mathcal{S} = \left\{ \left(\begin{smallmatrix}s_1\\ \vdots \\ s_n\end{smallmatrix}\right) \mid s_i \in \mathbb{F}_2 \right\}$. מתקיים כי $|\mathcal{S}| = 2^n$, לכן הפתרון הנאיבי של מעבר על כל ההשמות האפשריות הוא יקר מדי.

נגדיר $Y : \mathcal{S} \to \mathbb{R}^+$ על ידי $Y(S) =$מספר המשוואות ש-S מספקת. נרצה למצוא מקסימום של Y על $\mathcal{S}$. מהו $Y(S)$? נסמן ב-$r_1, \ldots, r_k \in \mathbb{F}_2^n$ את שורות המטריצה A ונסמן ב-$c_1, \ldots, c_n \in \mathbb{F}_2^k$ את עמודות המטריצה A אזי

$$\begin{aligned} Y(S) &= |\{i \in [k] : \langle r_i, s\rangle = b_i\}| = |\{i \in [k] : (As)_i = b_i\}| \\ &= \left| \left\{ i \in [k] \mid \left(\sum_{j=1}^{n} s_j c_j \right)_i = b_i \right\} \right| \end{aligned}$$

אלה הצגות שונות של $Y(S)$.

19.0.0.1 סיפור מסגרת - צפנים לתיקון שגיאות נרצה להעביר מידע (סדרת ביטים) בין A ל-B כאשר ביניהם יש ערוץ רועש. כאשר סדרת הביטים נכנסת לרעש הוא משנה אותה בהסתברות p. A מעביר הודעה m ל-B וערוץ הרעש מחזיר $\tilde{m}$ ל-B. כדי לתקן שגיאות שיוצר ערוץ הרעש, A מסתמש במטריצה K ומפעיל אותה על ההודעה שלו, כאשר B יודע על K. כלומר

$$A \to Km \to \text{NoisyChannel} \to \tilde{m} \to B$$

ואז מה שנותר ל-B הוא למצוא את ההודעה m המקיימת $Km = \tilde{m}$, כדי לעשות זאת הוא צריך למצוא פתרון שיקיים כמה שיותר משוואות, כמו בבעיה שלנו.

ננסה לבנות אלגוריתם דטרמיניסטי איטרטיבי שמכניס בכל איטרציה ערך למשתנה נוסף בצורה חמדנית.

דוגמה. נביט במערכת $\left(\begin{smallmatrix}1\\1\\1\end{smallmatrix}\right) = \left(\begin{smallmatrix}x_1\\x_2\\x_3\end{smallmatrix}\right) \left(\begin{smallmatrix}1&0&0\\0&1&1\\1&1&1\end{smallmatrix}\right)$ מערכת זו שקולה למערכת $x_1 \left(\begin{smallmatrix}1\\0\\1\end{smallmatrix}\right) + x_2 \left(\begin{smallmatrix}0\\1\\1\end{smallmatrix}\right) + x_3 \left(\begin{smallmatrix}0\\1\\1\end{smallmatrix}\right) = \left(\begin{smallmatrix}1\\1\\1\end{smallmatrix}\right)$. ננסה להבין מהו העקרון החמדני.

נציב $x_1 = 0$ ונקבל את המערכת $x_2 \left(\begin{smallmatrix}0\\1\\1\end{smallmatrix}\right) + x_3 \left(\begin{smallmatrix}0\\1\\1\end{smallmatrix}\right) = \left(\begin{smallmatrix}1\\1\\1\end{smallmatrix}\right)$ כלומר $\left(\begin{smallmatrix}0&0\\1&1\\1&1\end{smallmatrix}\right) \left(\begin{smallmatrix}x_1\\x_2\end{smallmatrix}\right) = \left(\begin{smallmatrix}1\\1\\1\end{smallmatrix}\right)$.

במקרה השני $x_1 = 1$ אנו מקבלים $1 \cdot \left(\begin{smallmatrix}1\\0\\1\end{smallmatrix}\right) + x_2 \left(\begin{smallmatrix}0\\1\\1\end{smallmatrix}\right) + x_3 \left(\begin{smallmatrix}0\\1\\1\end{smallmatrix}\right) = \left(\begin{smallmatrix}1\\1\\1\end{smallmatrix}\right)$ כלומר אנו מקבלים את המערכת $x_2 \left(\begin{smallmatrix}0\\1\\1\end{smallmatrix}\right) + x_3 \left(\begin{smallmatrix}0\\1\\1\end{smallmatrix}\right) = \left(\begin{smallmatrix}0\\1\\0\end{smallmatrix}\right)$.
במילים אחרות, כל חלוקה למקרים של משתנה מתפצלת לשתי מערכות קטנות יותר.

גישה הסתברותית תאפשר לנו לבנות כלל החלטה חמדני לבחירה בין שתי מערכות משוואות.

נגדיר מבנה של מרחב הסתברות על $\mathcal{S} = \mathbb{F}_2^n$ באופן הבא. לכל השמה $S \in \mathcal{S}$ ניתן הסתברות של $\frac{1}{2^n}$, אזי $\mathcal{S}$ הופך למרחב הססתברות והפונקציה $Y : \mathcal{S} \to \mathbb{R}^+$ הופכת למשתנה מקרי. נרצה לחשב את $\mathbb{E}(Y)$. בהנתן תוחלת זו עבור שתי המערכות שנקבל על ידי חלוקה למקרים של x_i נבחן את המערכת עם התוחלת הגבוהה היותר.

התוחלת ברוב המקרים היא $\frac{1}{2}$ מההסבר שאמרנו קודם, אבל יש מקרה מנוון בו היא 0, כאשר כל המשתנים הם אפס, או 1 אם הכל אפס.

למה. $\mathbb{E}\left(Y\right)=\begin{cases}1 & r=0,b=0\\ 0 & r=0,b=1\\ \frac{1}{2} & r\neq0\end{cases}$.

הוכחה. נחלק למקרים.

אם $r=0,b=0$ לכל השמה $S\in\mathbb{F}_2^n$ מתקיים כי $b=0=\langle r,s\rangle$ ולכן $Y\left(s\right)=1$.

אם $r=0,b=1$ אזי לכל השמה $S\in\mathbb{F}_2^n$ מתקיים כי $b\neq0=\langle r,s\rangle$ ולכן $Y\left(s\right)=0$.

אם $r\neq0$, נגדיר $V=\left\{ s:\langle r,s\rangle=0\right\}$ אזי V הוא תת מרחב וקטורי של $\mathbb{F}_2^n$. ממשפט המימדים נובע כי $\dim V+1=n$ ולכן $\dim V=n-1$ ולכן $|V|=2^{n-1}$.

אם $b=0$ אזי $Y\left(s\right)=1$ אם"ם $\langle r,s\rangle=0$ כלומר אם"ם $s\in V$ ולכן $\Pr\left(Y=1\right)=\frac{|V|}{2^n}=\frac{1}{2}$ ולכן $\mathbb{E}\left(Y\right)=1\cdot\Pr\left(Y=1\right)=\frac{1}{2}$.

אחרת, $b=1$ ובאופן דומה, $Y\left(s\right)=1$ אם"ם $s\in V^c$ ולכן $\Pr\left(Y=1\right)=\frac{|V^c|}{2^n}=\frac{1}{2}$ ושוב מקבלים $\mathbb{E}\left(Y\right)=\frac{1}{2}$. ■

הגדרה. תהי $Ax=b$ מערכת ליניאריות עם k משוואות. תהיינה $r_1,r_2,\ldots,r_k$ השורות של A. נאמר כי המשוואה ה-i עבור $1\leq i\leq k$ היא משוואה <u>ריקה טובה</u> אם $r_i=\vec{0}$ והסקלר $b_i=0$. נאמר כי i משוואה <u>ריקה רעה</u> אם $r_i=\vec{0}$ והסקלר $b_i=1$.

למה. (מרכזית) תהי $Ax=b$ מערכת משוואות ליניאריות עם k משוואות מעל $\mathbb{F}_2$. יהי β מספר המשוואות הריקות הרעות במערכת ו-γ מספר המשוואות הריקות הטובות. יהי Y המשתנה המקרי על $\mathcal{S}$ אשר ערכו בכל השמה s זה מספר המשוואות ש-s מספקת. אזי $\mathbb{E}\left(Y\right)=\frac{k+\gamma-\beta}{2}$.

הוכחה. עבור $1\leq i\leq k$ יהי Y_i המשתנה המקרי האחראי על המשוואה ה-i. אזי $Y_i\left(s\right)=\begin{cases}1 & \langle r_i,s_i\rangle=b_i\\ 0 & \langle r_i,s_i\rangle\neq b_i\end{cases}$ אזי $Y=\sum_{i=1}^{k}\mathbb{E}\left(Y_i\right)$

ולכן מלינאריות התוחלת

$$\mathbb{E}\left(Y\right)=1\cdot\gamma+0\cdot\beta+\frac{1}{2}\left(k-\beta-\gamma\right)=\frac{k+\gamma-\beta}{2}$$

כרצוי. ■

מסקנה. תהי $Ax=b$ מערכת של משוואות ליניאריות עם k משוואות ו-n נעלמים. אזי ניתן לחשב את התוחלת של Y בזמן $\mathcal{O}\left(k\cdot n\right)$.

הוכחה. לפי הלמה המרכזית, כדי לחשב את התוחלת, מספיק למצוא את מספרי המשוואות הריקות הטובות והרעות במערכת. לשך כך, נעבור על המטריצה A ונחפש שורות אפסים. אם מצאנו שורה כזו, נבדוק את ערך הווקטור b במקום המתאים כדי לוודא אם מצאנו משוואה ריקה טובה או רעה. ■

מסקנה. יהי $s^\star$ פתרון אופטימלי לבעיה אזי $\mathbb{E}\left(Y\right)\geq\frac{1}{2}Y\left(s^\star\right)$.

הוכחה. יהי β מספר המשוואות הריקות הרעות במערכת ו-γ מספר המשוואות הריקות הטובות. מכיוון שמשוואה ריקה רעה לא מסתפקת על ידי אף השמה ובפרט לא על ידי $s^\star$, מתקיים $Y\left(s^\star\right)\leq k-\beta$ ולכן לפי הלמה המרכזית

$$\mathbb{E}\left(Y\right)=\frac{k+\gamma-\beta}{2}\geq\frac{k-\beta}{2}\geq\frac{1}{2}Y\left(s^\star\right)$$

כרצוי. ■ נרצה להבין איך האלגוריתם שלנו יעבוד. נחזור לדוגמא שלנו:

דוגמה. נביט במערכת $\underbrace{\begin{pmatrix}1&0&0\\0&1&1\\1&1&1\end{pmatrix}\begin{pmatrix}x_1\\x_2\\x_3\end{pmatrix}=\begin{pmatrix}1\\1\\1\end{pmatrix}}_{\mathbb{E}(Y)=\frac{3}{2}}$ עם שתי המערכות $\underbrace{x_2\begin{pmatrix}0\\1\\1\end{pmatrix}+x_3\begin{pmatrix}0\\1\\1\end{pmatrix}=\begin{pmatrix}1\\1\\1\end{pmatrix}}_{\mathbb{E}(Y)=\frac{2}{2}=1},\underbrace{x_2\begin{pmatrix}0\\1\\1\end{pmatrix}+x_3\begin{pmatrix}0\\1\\1\end{pmatrix}=\begin{pmatrix}0\\1\\0\end{pmatrix}}_{\mathbb{E}(Y)=\frac{3+1}{2}=2}$ לכן נעדיף את המערכת השמאלית. נסתכל על מערכת זו, הממוצע של מספר המשוואות שמסתפקות הוא 2, והפתרון האופטימלי הוא 2,

לכן כל השמה אופטימלית כי פונקציית הערך קבועה. מלבד זאת, נבחין כי הממוצע של שתי התוחלת שקיבלנו הוא $\frac{3}{2} = \frac{2+1}{2}$ שזה התוחלת של המערכת הכללית. זה לא מקרי.

נסתכל על מרחב כל ההשמות $\mathcal{S}$. המרחב שלנו מחלק את המרחב לשני תת מרחבים באותו גודל - $s_1 = 0$ ו-$s_1 = 1$. לכל אחד מהחלקים מתאימה מערכת. אנו בוחרים את החצי עם התוחלת הגדולה יותר.

מהלמה המרכזית אנו מסיקים כי הממוצע על כל המרחב גדול מחצי המקסימום, ואילו הממוצע על החצי שבחרנו גדול יותר מהממוצע על הכל, ולכן גדול מחצי המקסימום הגלובלי בפרט.

באיטרציה הבאה אנו מחלקים את תת המרחב לשניים שוב, $s_2 = 0$ ו-$s_1 = 1$. נבחר את החצי החדש (עכשיו זה רביע מכל המרחב), הפעם הממוצע על החצי שבחרנו גדול מחצי המקסימום הכולל שוב, שכן הוא גדול מהממוצע על כלל החצי. נחזור על התהליך כאשר בשלב מסוים נקבל פתרון, יחיד המהווה חצי של תת מרחב, שתוחלתו גדולה יותר מחצי מהמקסימום ולכן זה יהיה פתרון 2 מקרב.

סימון תהי $\left(A^i, b^i\right)$ מערכת של משוואות לינאריות ב-m נעלמים. תהי Y פונקציית האיכות המתאימה למערכת זו, נסמן $Av\left(A^i, b^i\right) = \mathbb{E}\left(Y\right) = \frac{1}{2^m}\sum\limits_{s\in\mathbb{F}_2^m} Y\left(s\right)$ ממוצע הפתרונות לבעיה.

בצורה לא פורמלית, יהיו $y_1, \ldots, y_m$ המשתנים במערכת $\left(A^1, b^1\right)$. הצבת ערך 0 או 1 ל-y_1 מפצלת את המערכת לשתי מערכות אפשריות $\left(\tilde{A}, b_1\right), \left(\tilde{A}, b_0\right)$. העקרון החמדני יהיה שאם $Av\left(\tilde{A}, b_1\right) \geq Av\left(\tilde{A}, b_0\right)$ נציב $y_1 = 0$, אחרת נציב 1. נפרמל את מה שאמרנו עד כה.

למה. תהי $\left(A', b'\right)$ מערכת של משוואות לינאריות עם k משוואות ו-m נעלמים. יהי $y = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$ וקטור המשתנים של המערכת. הצבת ערך $s \in \{0,1\}$ במשתנה y_1 הופך את המערכת ל-$\left(\tilde{A}, \tilde{b}\right)$ כאשר $\tilde{A}$ מתקבלת מ-A' על ידי מחיקת העמודה הראשונה c_1 של A' ו-$\tilde{b} = b' - s \cdot c_1$.

הוכחה: המערכת $\left(A', b'\right)$ ניתנת לכתיבה הבאה. תהיינה $c_1, c_2, \ldots, c_m$ העמודות של A' אזי המערכת היא

$$y_1 \cdot c_1 + y_2 \cdot c_2 + \ldots + y_m \cdot c_m = b'$$

הצבת ערך s ב-y_1 הופכת את המערכת ל-

$$s \cdot c_1 + y_2 \cdot c_2 + \ldots + y_m \cdot c_m = b'$$

כלומר

$$y_2 \cdot c_2 + \ldots + y_m \cdot c_m = b' - s \cdot c_1$$

מה שמתאים למערכת $\left(\tilde{A}, \tilde{b}\right)$ בה $\tilde{A}$ מתקבלת מ-A' על ידי מחיקת העמודה הראשונה ו-$\tilde{b} = b' - s \cdot c_1$. ■

מסקנה. כאשר מציבים ערכים 0 ו-1 ב-y_1 מקבלים שתי מערכות $\left(\tilde{A}, b_0\right)$ כאשר $b_0 = b'$ ו-$\left(\tilde{A}, b_1\right)$ כאשר $b_1 = b' - c$ ו-$\tilde{A}$ מתקבלת מ-A' על ידי מחיקת העמודה הראשונה.

למה. (הפרדת הממוצע לממוצע תתי המרחבים) $Av\left(A', b'\right) = \frac{Av\left(\tilde{A}, b_0\right) + Av\left(\tilde{A}, b_1\right)}{2}$.

הוכחה: יהי Y פונקציית האיכות של המערכת (A',b') אזי $Av\left(\tilde{A},b_0\right)=\frac{1}{2^{m-1}}\sum_{s\in\mathbb{F}_2^m,s_1=0}Y(s)$ כנ"ל לגבי $Av\left(\tilde{A},b_1\right)$. לכן

$$\begin{aligned}\frac{Av\left(\tilde{A},b_0\right)+Av\left(A,b_1\right)}{2}&=\frac{\frac{1}{2^{m-1}}\sum_{s\in\mathbb{F}_2^m,s_1=0}Y(s)+\frac{1}{2^{m-1}}\sum_{s\in\mathbb{F}_2^m,s_1=1}Y(s)}{2}\\&=\frac{1}{2^m}\left(\sum_{s\in\mathbb{F}_2^m,s_1=0}Y(s)+\sum_{s\in\mathbb{F}_2^m,s_1=1}Y(s)\right)\\&=\frac{1}{2^m}\left(\sum_{s\in\mathbb{F}_2^m}Y(s)\right)=Av\left(A',b'\right)\end{aligned}$$

כרצוי. ■ נותר לנו רק להציג את האלגוריתם בצורה פורמלית.

19.1 אלגוריתם 2 מקרב לבעיה

קלט מטריצה A עם k שורות ו-n עמודות מעל $\mathbb{F}_2$.

פלט השמה $s\in\mathbb{F}_2^n$ כך ש-$Y(s)\geq\frac{1}{2}Y(s^\star)$ כאשר $s^\star$ פתרון אופטימלי.

אלגוריתם 18 אלגוריתם 2 מקרב לבעייה

---

1. <u>אתחול</u> נאתחל $A'=A$ ו-$b'=b$
2. <u>איטרציה</u>: יהי $0\leq t\leq n-1$. באיטרציה ה-$t+1$ של האלגוריתם נציב ערך במשתנה x_{t+1} אחרי שהגדרנו כבר ערכים $x_t\sim s_t,\ldots,x_1\sim x_1$. ההצבה של הערכים $0,1$ למשתנה x_{t+1} מפצלת את המערכת (A',b') לשתי מערכות $\left(\tilde{A},b_0\right),\left(\tilde{A},b_1\right)$ כפי שתיארנו במסקנה הקודמת. אם $Av\left(\tilde{A},b_0\right)\geq Av\left(\tilde{A},b_1\right)$ נגדיר $s_{t+1}=0$ אחרת נגדיר $s_{t+1}=1$. נציב $x_{t+1}=s_{t+1}$. נעדכן $A'=\tilde{A}$ ו-$b'=b_{s_{t+1}}$. נעדכן $t=t+1$.
3. <u>עצירה</u>: כאשר $t=n$ נעצור ונחזיר את s.

משפט. האלגוריתם הנ"ל הוא 2 מקרב כלומר $Y(s)\geq\frac{1}{2}Y(s^\star)$.

הוכחה: עבור $0\leq t\leq n$ נסמן ב-(A_t,b_t) את מערכת המשוואות אחרי t האיטרציות הראשונות של האלגוריתם. נסמן $E_t=Av(A_t,b_t)$ נוכיח כי לכל $0\leq t\leq n-1$ מתקיים $E_{t+1}\geq E_t$. אם נוכיח זאת נסיים. כדי לראות זאת נשים לב כי

$$E_t=\frac{1}{2^{n-t}}\sum_{\substack{r\in\mathbb{F}_2^n\\r_1=s_1,\ldots,r_t=s_t}}Y(r)$$

בפרט $E_0=Av(A,b)$ וראינו כי $Av(A,b)\geq\frac{1}{2}Y(s^\star)$ ו-$E_n=Y(s)$ וכן $E_n=Y(s)$. לכן נקבל $Y(s)=E_n\geq E_{n-1}\geq\ldots\geq E_0=Av(A,b)\geq\frac{1}{2}Y(s^\star)$

נקבע $0\leq t\leq n-1$ ונוכיח כי $E_{t+1}\geq E_t$. נסמן ב-$\left(\tilde{A},b_1\right),\left(\tilde{A},b_0\right)$ את שתי המערכות המתקבלות לאחר ההצבה של $0,1$ למשתנה x_{t+1} באיטרציה ה-$t+1$ של האלגוריתם. אזי לפי העקרון החמדני של האלגוריתם נקבל

$$E_{t+1}=Av(A_{t+1},b_{t+1})=\max\left\{Av\left(\tilde{A},b_0\right),Av\left(\tilde{A},b_1\right)\right\}\geq\frac{Av\left(\tilde{A},b_0\right)+Av\left(\tilde{A},b_1\right)}{2}\overset{\text{למה 2}}{=}Av(A_t,b_t)=E_t$$

כרצוי. ■

20 תרגול 8 - אלגוריתמי קירוב, בעיית MaxCut, בעיית הסוכן הנוסע המטרי, רשתות זרימה

20.1 בעיית MaxCut

ראשית נגדיר מהו חתך.

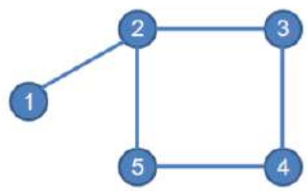
הגדרה. יהי $G = (V, E)$ גרף לא מכוון. חתך בגרף הוא חלוקה של הקודקודים לשתי קבוצות זרות A, B כך ש-$V = A \cup B$ ו-$A \cap B = \emptyset$. נסמן ב-E_c קבוצת הצלעות בחתך $E_c = \{(u, v) \in E \mid (u \in A \wedge v \in B) \vee (u \in B \vee v \in A)\}$.

נגדיר אם כך את הבעיה.

קלט גרף $G = (V, E)$

פלט חתך בגרף בגודל מקסימלי (עם קבוצת צלעות מקסימלית).

דוגמה. נביט בגרף הבא:



איור 5: הגרף בדוגמא

נרשום חתכים אפשריים:

A	B	E_c
$\{1, 2, 3, 4, 5\}$	$\emptyset$	$\emptyset$
$\{1, 2\}$	$\{3, 4, 5\}$	$\{(2, 3), (2, 5)\}$
$\{1, 3, 5\}$	$\{2, 4\}$	כל הצלעות

מכאן נסיק כי החתך $E_{\{1,3,5\} \cup \{2,4\}}$ הוא מקסימלי.

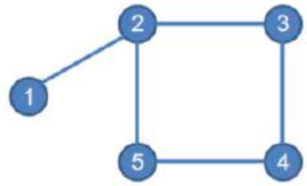
הבעיה היא כמובן NP קשה, ונרצה להציע לה פתרון מקרב כלשהו. מה נעשה אם כך? נרצה שבכל הוספה של קודקוד לקבוצה אחת, יהיו לו כמה שיותר שכנים בקבוצה השנייה. נציע את האלגוריתם ה-2 מקרב הבא.

אלגוריתם 19 אלגוריתם 2 מקרב ל-MaxCut

---

1. נמספר את הקודקודים $V = \{v_1, \ldots, v_n\}$
2. נאתחל $A = V, B = \emptyset$.
3. נעבור על הקודקודים לפי הסדר:
   (א) אם מספר השכנים של הקודקוד הנוכחי בקבוצה שלו גדול ממספר השכנים שלו בקבוצה השנייה - נעביר אותו קבוצה.
4. נחזור על שלב 3 עד שלא ישארו קודקודים שצריך להעביר.
5. נחזיר את A, B.

דוגמה. נריץ את האלגוריתם על הדוגמא הבאה:



איור 6: קלט דוגמא לאלגוריתם

כולם כרגע באותה קבוצה, אם כך, נעביר את 1 לקבוצה השנייה. עתה, ל-2 יש שני שכנים בקבוצה A ולכן עדיף להעביר אותו ל-B. 3 יבעל מספר זהה של שכנים ולכן נשאיר אותו ב-A, 4 בעל שני שכנים ב-A ולכן יועבר ל-B ו-5 ישאר לבן כי יש לו שני שכנים מ-B כלומר קיבלנו

$$A = \{3,5\}\,, B = \{1,2,4\}$$

באיטרציה הראשונה. עכשיו אנחנו חוזרים על התהליך. אנו רואים כי ל-1 יש שכן ב-B לכן נעביר אותו קבוצה, שאר הקודקודים נשארים באותו מקום ולכן אנו מקבלים $A = \{1,3,5\}\,, B = \{2,4\}$. זה הפלט של האלגוריתם.

משפט. האלגוריתם הוא 2 מקרב לבעיית MaxCut.

הוכחה: נוכיח חוקיות, קירוב ועצירה.

עצירה: נבחין כי בכל איטרציה אנו מגדילים את מספר הצלעות בחתך, שכן אנו מעבירים קודקודים מקבוצה לקבוצה אם"ם זה מגדיל את מספר הצלעות בחתך. כיוון שמספר הצלעות בחתך חסום על ידי $|E|$, אנו מבצעים לכל היותר על ידי $|E|$ איטרציות.

חוקיות: נוכיח כי $A\cap B=\emptyset$. אתחלנו את A,B כך ש-$A[]\dot{\cup}B=V$. בכל איטרציה העברנו קודקודים מקבוצה לקבוצה, בלי שכפול, הסרה וכו' לכן A,B נשארו קבוצות זרות שמקיימות $V=A\cup B$.

קירוב: נראה כי עבור החתך המחוזר $C=(A,B)$ מתקיים כי $|E_c|\geq\frac{1}{2}\mathrm{OPT}$. נסמן ב-$E_{v,C}$ צלעות בחתך שנוגעות בקודקוד v. אזי מספר השכנים בקבוצה השנייה גדול ממספר השכנים בקבוצה של v, ולכן $|E_{v,C}|\geq\frac{1}{2}\deg v$ואזי

$$\begin{aligned}|E_c| &= \frac{1}{2}\sum_{v\in V}|E_{v,C}| \geq \frac{1}{2}\sum_{v\in V}\frac{1}{2}\deg v\\ &\geq \frac{1}{4}\sum_{v\in V}\deg v = \frac{1}{2}|E| \geq \frac{1}{2}\mathrm{OPT}\end{aligned}$$

כרצוי. ■ ננתח את זמן הריצה של האלגוריתם.

- אתחול: $\mathcal{O}(V)$.
- איטרציה: עוברים על כל הקודקודים ובדיקת קבוצה השייכות של כל קודקוד היא $\mathcal{O}(|V|)$ לכן סך הכל $\mathcal{O}\left(|V|^2\right)$
- חוזרים על שלב האיטרציה לכל היותר $|E|$ פעמים ולכן זמן הריצה הוא $\mathcal{O}\left(|E|\,|V|^2\right)$.

20.2 בעיית הסוכן הנוסע המטרי (M-TSM)

20.2.0.1 סיפור כיסוי אנחנו סוכן מכירות שרוצה לטייל בעולם ולמכור את המוצרים שלנו. נרצה למזער את עלות הטיסות שלנו ככה שנגיע לכל היעדים.

קלט גרף מלא $G=(V,E)$ ופונקצית משקל $w:E\to\mathbb{R}^+$.

פלט מעגל פשוט העובר בכל קודקודי הגרף - מעגל המילטוני $C=(e_1,\ldots,e_n)$ עם משקל מינימלי.

זו בעיה NP קשה, ולכן עלינו להניח הנחה מקלה.

הנחה מקלה אי שוויון המשולש מתקיים עבור w, כלומר ו$\forall x, y, z \in V : w\left(x, z\right) \leq w\left(x, y\right) + w\left(y, z\right)$.

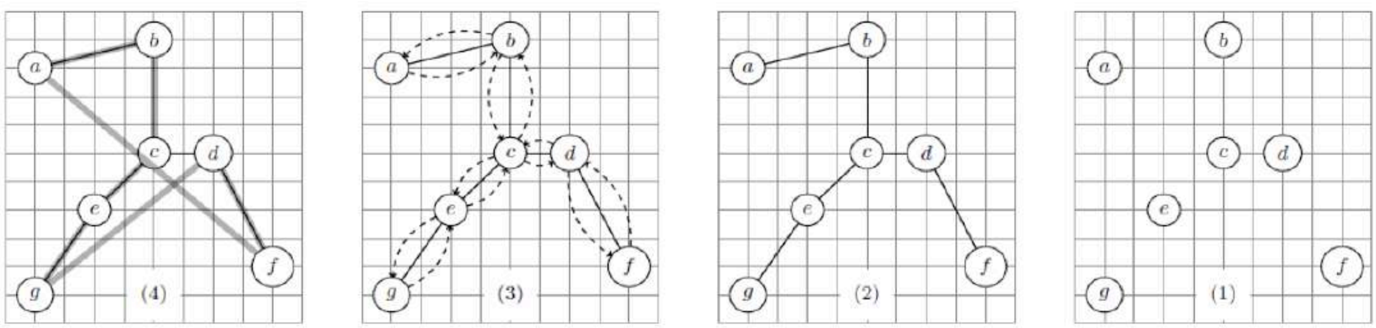
לפני שניגש לאלגוריתם, ניתן קצת אינטואיציה. אנחנו רוצים למלא מעגל פשוט בעל משקל מינימלי, אם נוציא ממנו צלע נקבל עץ פורש, ננסה להעזר בעצים פורשים מינימלי כדי להשיג מעגל פשוט זה, שכן עץ פורש מינימלי אנו יודעים למצוא. נציע את האלגוריתם ה-2 מקרב הבא:

אלגוריתם 20 אלגוריתם 2 מקרב ל-M-TSM

1. נמצא עץ פורש מינימלי בגרף MST שנסמנו T.
2. נאתחל $H = \emptyset$ - המעגל ההמילטוני שנחזיר.
3. נריץ על הגרף DFS:
   (א) נבחר קודקוד אקראי v_i ונריץ עליו DFS על T.
   (ב) בכל פעם שנבקר בקודקוד בפעם הראשונה נוסיף אותו ל-H.
   (ג) בסיום הרצת ה-DFS נוסיף את v_i ל-H.
4. נחזיר את H.

כאן שלב ה-DFS נועד לסרוק את העץ הפורש שלנו לעומק.

דוגמה. ניתן דוגמת הרצה לאלגוריתם:



איור 7: קלט דוגמא לאלגוריתם

נתחיל מ-$v = a$, נקבל מהרצת ה-DFS את $H = \{a, b, c, e, g, d, f, a\}$ ואת הצלעות $\{(a,b), (b,c), (c,e), (e,g), (g,d), (d,f), (f,a)\}$.
למעשה, אנו רואים מכאן כי עד שיצאנו מהעץ הפורש, קיבלנו משקל מינימלי, ה-TradeOff הוא מהצלעות שלא בעץ הפורש המינימלי, דהיינו $(f,a), (g,d)$. נבחין כי נוכל להגדיל את מספר הצלעות על ידי הוספת קודקוד בכל פעם שהגענו אליו ב-DFS כלומר במקרה זה נקבל הליכה F המוגדרת על ידי $F = \{a, b, c, e, g, e, c, d, f, d, c, b, a\}$, כל צלע שנקבל מופיעה בעץ הפורש לכל היותר פעמיים ולכן $w(F) \leq 2 \cdot w(T)$, אבחנה זו תהיה אלמנט מרכזי בהוכחה. יתר על כן, מההנחה המקלה, אנו מקבלים כי $w(F) \geq w(H)$, שכן

$$w(g,e) + w(e,c) + w(c,d) \overset{\triangle}{\geq} w(g,c) + w(c,d) \overset{\triangle}{\geq} w(g,d)$$

כלומר כל רצף צלעות שהוספנו גדול מהצלע היחידה שהוספנו בפלט האלגוריתם, נוכיח במפורט בהוכחה.

משפט. האלגוריתם מחזיר פתרון 2-מקרב לבעיית ה-M-TSM.

הוכחה: נוכיח חוקיות וקירוב.

<u>חוקיות</u>: מהגדרת DFS אנו עוברים על כל הקודקודים ומוסיפים קודקוד רק אם הוא לא נמצא ב-H, לכן כל קודקוד נמצא ב-A בדיוק פעם אחת, פרט לקודקוד הראשון, שאותו אנו מוסיפים בסיום ריצת DFS. G הוא גרף מלא ולכן לכל שני קודקודים רצופים ב-H קיימת צלע.

<u>קירוב</u>: נסמן $H^\star = \mathrm{OPT}$. נראה כי $w(H) \leq 2 \cdot w(H^\star)$. נעזר בכך שמצאנו עץ פורש מינימלי T. תחילה נראה כי $w(T) \leq w(H^\star)$ ואז נראה כי $w(H) \leq 2w(T)$.

נסיר צלע e כלשהי מ-$H^\star$ ונקבל עץ פורש שנסמנו $T^\star$, מתקיים כי $w(T^\star) = w(H^\star) - w(e)$ ולכן $w(T) \leq w(T^\star) \leq w(H^\star)$.

נגדיר הליכה על הגרף באופן הבא. נגדיר F להיות האוסף הנוצר על ידי הרצת ה-DFS על הגרף והוספת כל קודקוד שמגיעים אליו ל-F, כאשר יש כפילויות ב-F. נשים לב כי H מתקבלת מ-F על ידי הסדרת הקודקודים.

למה. יהא C מסלול פשוט בגרף ויהא C' מסלול המתקבל מ-C על ידי החסרת קודקודיהם ואיחוי בסדר המסלול, אזי מתקיים $w(C') \leq w(C)$.

הוכחה: נסמן $x, y, z \in V$ קודקודים ב-C לפי סדר רצוף זה במסלול, ו-C' מתקבלת מ-C על ידי הסרת הקודקוד y ואיחוי בין החלקים החדשים באותו הסדר. אזי $C' = (C \setminus \{(x,y),(y,z)\}) \cup \{x,z\}$. אזי

$$\begin{aligned} w(C') &= w(C) - w((x,y)) - w((y,z)) + w(x,z) \\ &\overset{\triangle}{\leq} w(C) \end{aligned}$$

כרצוי. ■ מכאן אנו מקבלים כי $w(H) \leq w(F)$ אבל כל צלע שמתקבלת מ-F חוזרת לכל היותר פעמיים ולכן

$$w(H) \leq w(F) \leq 2 \cdot w(T) \leq 2 \cdot w(H^\star)$$

כמו שרצינו. ■ ננתח את זמן הריצה.

- <u>אתחול</u>: מציאת עץ פורש מינימלי אפשר לבצע ב-$\mathcal{O}\left(|V|^2\right)$, עם אלגוריתם שעוד לא ראינו.
- <u>הרצת DFS</u>: $\mathcal{O}(|V| + |E|)$.

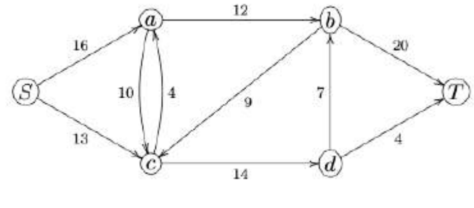
סך הכל קיבלנו $\mathcal{O}\left(|V|^2\right) = \mathcal{O}\left(|V|^2 + |V| + |E|\right)$.

20.3 רשתות זרימה

20.3.0.1 סיפור רקע נתונה לנו רשת צינורות בתצורת גרף מכוון בו לכל צלע יש קיבולת שהיא מספר אי שלילי. בגרף שני קודקודים מיוחדים - קודקוד s שמייצר חומר וקודקוד t שסופק חומר. מטרתנו היא להעביר מ-s ל-t כמה שיותר חומר כאשר המגבלה שלנו היא שלא ניתן להזרים בצינור יותר מהקיבולת שלו, ולכל קודקוד, פרט ל-s ול-t, כמות החומר שנכנסת אליו צריכה גם לצאת ממנו.

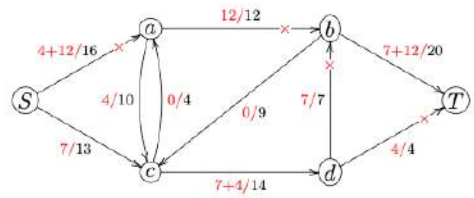
לבעיות מסוג זה אנו קוראים "בעיות זרימה". אפשר למדל בעיות רבות באמצעות זרימה ברשתות. למשל, העברית מים בצינורות, העברת ייצור ממפעל ליישוב דרך כמה קווי שינוע והעברת מידע באינטרנט.

דוגמה. נביט בגרף הבאה להמחשה:



איור 8: המחשה לרשת זרימה

נוכל להגדיר עליה את הזרימה הבאה, באדום:



איור 9: אנו מעבירים 23 יחידות חומר מ-s ל-t, וסימנו כל צינור המגיע לרוויה ב-$\times$. נשים לב שהפונקציה הזו אכן מקיימת את כל התכונות שציפינו להן. האם ניתן להזרים יותר ברשת זו? לא, שכן כל מסילה מ-s ל-t עוברת דרך אחת מהצלעות המסומנות ב-$\times$ ואנו מעבירים את המקסימום שניתן דרך צלעות אלו, המהוות צוואר בקבוק.

נגדיר את המושגים באופן פורמלי.

הגדרה. רשת זרימה היא חמישיה (V,E,c,s,t) כאשר:

- $G=(V,E)$ גרף מכוון.
- $c:E\to\mathbb{R}^+$ פונקציית קיבול (Capacity).
- $s\in V$ קודקוד מקור (Source).
- $t\in V$ קודקוד מטרה/בור/כיור (Sink/Target).

הנחה אנו מניחים שאין צלעות שיוצאות מ-t או צלעות הנכנסות ל-s.

הגדרה. זרימה חוקית ברשת זרימה היא פונקציה $f:E\to\mathbb{R}^+$ המקיימת שני אילוצים:

(i) : אילוץ הקיבול: $\forall e\in E: f(e)\le c(e)$.

(ii) : חוק שימור החומר: $\sum\limits_{u:(v,u)\in E} f(v,u) = \sum\limits_{u:(u,v)\in E} f(u,v)\ , \quad \forall v\in V\setminus\{s,t\}$.

הגדרה. השטף של הזרימה f - המסומן ב-$|f|$מוגדר כ-"סך כל הזרימה היוצאת מהמקור" כלומר $|f| := \sum\limits_{u:(s,u)\in E} f(s,u)$.

מסקנה. מחוק שימור החומר נובע כי $\sum\limits_{u:(u,t)\in E} f(u,t) = \sum\limits_{u:(s,u)\in E} f(s,u)$, כלומר הזרימה הנכנסת לבור זהה לזרימה היוצאת מהמקור.

20.3.1 בעיית הזרימה

קלט רשת זרימה $N=(V,E,c,s,t)$.

פלט זרימה חוקית f ברשת הזרימה N בעלת שטף $|f|$ מקסימלי.

נשים לב שזוהי בעיית אופטימיזציה, כאשר פתרון חוקי הוא איזשהי זרימה חוקית ברשת, ופתרון אופטימלי הוא זרימה חוקית בעלת שטף מקסימלי.

הערה. ייתכנו כמה זרימות מקסימליות ברשת נתונה, אבל השטף של כולן זהה.

הערה. בכל רשת יש לפחות זרימה חוקית אחת - זרימת האפס עם $f\equiv 0$.

הערה. בהמשך נראה שני אלגוריתמים לפתרון הבעיה, פורד-פאלקרסון הפתור אותה בזמן $\mathcal{O}(|E|\,|f^\star|)$ כאשר $f^\star$ הזרימה האופטימלית, והשיפור של אדמונד-קארפ הפותר אותה ב-$\mathcal{O}\left(|V|\,|E|^2\right)$. שני אלגוריתמים אלה יהוו קופסא שחורה המקבלת רשת זרימה ומחזירה זרימה חוקית מקסימלית, בדומה לקופסאות השחורות שלנו בתכנון לינארי, נציג את הבעיה בצורת רשת זרימה ואז נכניס אותה לקופסה השחורה.

חלק V

רשתות זרימה NetworkFlow

נגדיר בצורה אינטואיטיבית כמה מושגים.

הגדרה. רשת זרימה זה גרף מכוון עם קיבולם חיוביים על הצלעות עם שני קודקודים מיוחדים, קודקוד מקור ו-קודקוד הבור.

הגדרה. זרימה ברשת זרימה זו פונקציה מהצלעות למספרים אי-שליליים. המקיימת את האילוצים הבאים:

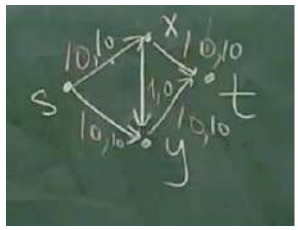
(i) : אילוץ הקיבול: לכל צלע בגרף הזרימה בצלע אינה גדולה מהקיבול של צלע.

(ii) : חוק שימור החומר: לכל קודקוד בגרף חוץ מקודקודי המקור והבור, הזרימה הנכנסת לקודקוד שווה לזרימה היוצאת מהקודקוד.

הגדרה. (עצמת הזרימה) שטף הזרימה מוגדר כזרימה הכוללת היוצאת מקודקוד המקור.

בעיה. בהנתן רשת זרימה, נרצה למצוא זרימה מקסימלית (זרימה עם שטף מירבי) ברשת זו.

דוגמה. יהיו 4 קודקודים x, t, y, s והגרף הבא:



איור 10: הזרימה המקסימלית היא 20 שכן נוכל להזרים 10 משתי הצלעות של s. לא נוכל להזרים יותר כי זה מקסימום על הקיבולים.

נגדיר עכשיו את המושגים פורמלית.

הגדרה. רשת זרימה זו חמישיה $N = (V, E, c, s, t)$ כאשר $G = (V, E)$ הוא גרף מכוון ו-$c : E \to \mathbb{R}_{>0}$ זו פונקציית הקיבול של הרשת ו-$s, t \in V$ הם קודקודים מיוחדים, s קודקוד המקור ו-t קודקוד הבור.

הערה. זה יהיה הקלט לבעיה שלנו.

הגדרה. זרימה ברשת זרימה N היא פונקציה $f : E \to \mathbb{R}_{\geq 0}$ המקיימת את שני האילוצים הבאים:

(i) : אילוץ הקיבול: לכל $e \in E$ מתקיים $0 \leq f(e) \leq c(e)$.

(ii) : חוק שימור החומר: לכל $x \in V \setminus \{s, t\}$ מתקיים $\underbrace{\sum_{\substack{w \in V \\ (x,w) \in E}} f(x, w)}_{\text{הזרימה היוצאת מ-}x} = \underbrace{\sum_{\substack{u \in V \\ (u,x) \in E}} f(u, x)}_{\text{הזרימה הנכנסת ל-}x}$.

הגדרה. שטף הזרימה f מוגדר כ-$|f| = \sum_{\substack{v \in V \\ (s,v) \in E}} f(s, v)$.

בעיה. בהנתן רשת זרימה N, נרצה למצוא זרימה חוקית f ברשת כך ש-$|f|$ מקסימלי.

ניתן לפתור זאת על ידי תכנון לינארי[3] היתרונות של $F\&F$ ו-$E\&K$ על תכנון לינארי הוא שזה יותר יעיל וגם אם הקיבולים מספרים שלמים, הזרימה המוחזרת על ידי $F\&F$ היא זרימה עם ערכים שלמים, בשונה מ-LP שמחזיר ערכים שלא בהכרח שלמים.

מרחב הפתרונות החוקיים לבעיה זו הוא מרחב כל הזרימות החוקיות ברשת N.

הערות

1. האם הוא תמיד לא ריק? כן. זרימת האפס $f(e) = 0$ לכל $e \in E$ היא זרימה חוקית

2. המרחב הזה יכול להיות אינסופי. נשים לב כי אם f, g שתי זרימות חוקיות אז גם $\frac{f+g}{2}$ היא זרימה חוקית. באופן כללי יותר, לכל $0 \leq \lambda \leq 1$ גם $h = \lambda f + (1 - \lambda) g$ גם היא זרימה חוקית ולכן המרחב של כל הזרימות החוקיות הוא קבוצה קמורה (בפרט בדרך כלל אינסופית).

(א) האם בכל רשת זרימה של 100 יש זרימה של 75? כן, כי נוכל לבחור $\lambda = \frac{75}{100} = 0.75$ ו-g פונקציית האפס, למעשה לכל $0 \leq a < 100$ יש זרימה חוקית.

[3] FordFulkerson מצאו לזה פתרון בשנת 1981 ללא תכנון לינארי. בהמשך זה נפתר עם אלגוריתם EdmondKorp

3. למרות שמרחב הפתרונות החוקיים הוא אינסופי, יש פתרון אופטימלי. נבחין כי מרחב הפתרונות הוא קבוצה סגורה וחסומה מעל $\mathbb{R}^{|\mathbb{E}|}$ ולכן זו קבוצה קומפקטית ופונקציית השטף היא רציפה על זרימות.

שאלה כיצד ניצור זרימה לא טריוויאלית בהנתן רשת זרימה?

תשובה נמצא (בעזרת BFS) מסלול מ-s ל-t ונשלח אליו זרימה קבועה, השווה לקיבול המינימלי של הצלעות במסלול. מחוץ למסלול נגדיר את הזרימה להיות אפס. ברשת הקודמת, היינו יכולים לבחור לשלוח 10 מ-s ל-x ל-t ובכל השאר אפס.

שאלה כיצד נוכל להגדיל את הזרימה?

תשובה נמצא עוד מסלול ונזרים עוד זרימה קבועה. ניקח את הממוצע של שתי הזרימות. ככה נמשיך עד שלא נוכל להזרים עוד. דבר זה מוביל לאלגוריתם האיטרטיבי הבא:

אלגוריתם 21 הצעה לאלגוריתם איטרטיבי למקסום זרימה ברשת זרימה

1. בכל שלב נמצא מסלול פשוט בין s ל-t. נשלח זרימה קבועה במסלול ונוסיף את הזרימה החדשה הזו לזרימה שקיימת כרגע ברשת תוך כדי כך שהוא דואג לשמור על אילוץ הקיבול.

נריץ אלגוריתם זה על הדוגמא שלנו. נתחיל מהרשת המקורית N_0. נמצא מסילה $p_0 : s \xrightarrow{10} x \xrightarrow{1} y \xrightarrow{10} t$ ונזרים בה 1. קיבלנו רשת חדשה עם אותם קודקודים רק שעכשיו כל צלע במסילה בעלת קיבול פחות הקיבול מהמסלול p_0 שהוא 1. נמצא מסילה חדשה $p_1, f_1 : s \xrightarrow{9} x \xrightarrow{10} t$ עם זרימה 9. נחסיר מצלעות המסלול מהגרף החדש את זרימה זו, נזרוק צלעות בעלות קיבול 0. קיבלנו רשת N_2 שבה הקיבולים האפקטיביים הם $c(s,y) = 10, c(y,t) = 9, c(x,t) = 1$. נותרה לנו עוד איטרציה אחת. נזרים 9 במסלול $s \xrightarrow{10} y \xrightarrow{9} t$. נזרוק צלעות בעלות קיבול 0 בגרף המתקבל לאחר הורדת הקיבולים של המסלול. קיבלנו אך ורק את הצלע $s \xrightarrow{1} y, x \xrightarrow{1} t$, זה גרף לא קשיר ולכן סיימנו. נקבל סך הכל קיבול $9 + 9 + 1 = 19$ לא אופטימלי. כלומר האלגוריתם שלנו לא אופטימלי. מדוע? נבחין כי הזרימה תלויה במסלול הראשון שאנחנו בוחרים ונרצה להיות מסוגלים לחזור אחורה ולהוריד זרימות במידה והבחירה איננה טובה. האלגוריתם שלנו לא עושה זאת, אלא בוחר כל פעם מסלול ומוסיף את הזרימה. זה מאוד נאיבי. יחד עם זאת, נראה אלגוריתם שמסתמך על הרעיון הבסיסי.

הנחות מקלות

לפני שנוכיח את הלמה נניח כמה דברים מקלות:

1. אין בגרף (V,E) של הרשת לולאות.
2. אין צלעות הנכנסות למקור.
3. אין צלעות היוצאות מהבור.

למה. לכל רשת זרימה קיימת זרימה אופטימלית f עם התכונה הבאה: לא קיימים שני קודקדים $x \neq y \in V$ כך ש-$(y,x), (x,y) \in E$ וכך ש-$f(x,y) > 0$ וגם $f(x,y) > 0$.

האינטואיציה היא שבמקום להזרים חזרה, ניקח את הזרימה המינימלית מבין השתיים ונחסיר אותה מהזרימה הגדולה.

הוכחה: (הלמה) תהי g זרימה אופטימלית ברשת. נגדיר $B = \{(x,y) \in E : (y,x) \in E \wedge g(x,y) > 0 \wedge g(y,x) > 0\}$. נשים לב כי $(x,y) \in E$ אם"ם $(y,x) \in B$.

נגדיר פונקציה חדשה f על E באופן הבא:

$$f(e) = \begin{cases} g(e) & e \notin B \\ g(x,y) - \min\{g(x,y), g(y,x)\} & (x,y) = e \in B \end{cases}$$

נוכיח כי f היא זרימה חוקית ברשת, כי $|f|=|g|$ ולכן f אופטימלית, וכי ל-f יש התכונה הנדרשת.

<u>חוקיות f</u>: נבחין כי f אי שלילית על פי הגדרתה. בנוסף, f מקיימת את אילוץ הקיבול כי לכל $e\in E$ מתקיים כי $f(e)\le g(e)\le c(e)$. יתר על כן, f מקיימת את חוק שימור החומר כי לכל $x\in V\setminus\{s,t\}$ במעבר מ-g ל-f הזרימה הנכנסת ל-x והזרימה היוצאת ממנו קטנות באותה המידה.

<u>אופטימליות</u>: נזכור כי על פי ההנחה שעשינו, אין צלעות הנכנסות לקודקוד המקור s זה אומר שלא קיים $v\in V$ כך ש-$(x,s)\in E$ ו-$(s,x)\in E$ ולכן לכל צלע $(s,x)\in E$ מתקיים $f(s,x)=g(s,x)$ ולכן $g(s,x)\underset{\substack{x\in V\\(s,x)\in E}}{\sum} f(s,x)=\underset{\substack{x\in V\\(s,x)\in E}}{\sum}=|f|$.

<u>קיום התכונה</u>: נשאר לוודא כי לא קיימים $x\neq y\in V$ כך ש-$(x,y)\in E,(y,x)\in E$ וכך ש-$f(x,y),f(y,x)>0$. תהי $(x,y)\in E$ כך ש-$(y,x)\in E$. אמנם אם $(x,y)\in B$ אזי על פי הגדרת f מתקיים $\min\{f(x,y),f(y,x)\}=0$ ואם $(x,y)\notin B$ אזי לא ייתכן כי $\min\{f(x,y),f(y,x)\}>0$, אחרת, מהיות $\min\{g(x,y),g(y,x)\}=0$ (שכן אחרת $(x,y)\in B$) נקבל כי $f>g$ בסתירה לכך ש-$f\le g$. ■ עולה השאלה, איך FordFulkerson תיקנו את האלגוריתם שהרצנו? בדוגמא הקודמת, בחרנו מסלול $p_0: s\xrightarrow{10}x\xrightarrow{1}y\xrightarrow{10}t$ והזרמנו בו 1. נרצה להיות מסוגלים ליצור זרימה חדשה, ללא הסתמכות על הגרף של הקיבולים האפקטיביים. כדי לעשות זאת, נאפשר להזרים זרימה נגדית כלומר זרימה מנוגדת לזרימה שהזרמנו קודם. ככה, נקבל שבמידה ואין לנו אפשרות להזרים דרך צלע מסויימת נוכל להזרים דרך הכיוון ההפוך שלה זרימה שלילית וכך לקבל אפשרות להזרים דרכה זרימה חיובית. הרשת שתתקבל עקב הוספת זרימות שליליות אלה תהיה רשת וירטואלית שתעזור לנו לבנות זרימה אופטימלית. למעשה אנו נרחיב את מושג הזרימה כך שהגרף יכיל את כל הצלעות בין הקודקודים, אבל צלע שלא קיימת בגרף המקורי תהיה בעלת קיבול 0 ויתכנו זרימות שליליות. נפתח מושג זה בהרחבה בפעם הבאה.

נשנה את הרשת שנעבוד איתה לאורך הריצה של האלגוריתם נחליף את קיבולי הרשת בקיבולים אפקטיביים כאשר הקיבול האפקטיבי של צלע הוא הקיבול המקורי פחות הזרימה כרגע בצלע, כאשר הקיבול האפקטיבי של צלע שווה ל-0 (הצלע רוויה) הצלע יורדת מהרשת ולכן הרשת עצמה משתנה וקטנה.

20.4 הרחבה של רשת זרימה

הגדרה. <u>רשת זרימה מורחבת</u> זו רביעיה $N'=(V,c',s,t)$ כאשר V קבוצת הקודקודים של הרשת, $s\in V$ הוא קודקוד המקור, $t\in V$ הוא קודקוד הבור ו-$c':V\times V\to\mathbb{R}_{\ge 0}$ היא פונקציית הקיבול המורחב המקיימת:

(i) : לכל $x\in V$ מתקיים כי $c'(x,x)=0$.

(ii) : לכל $x\in V$ מתקיים כי $c'(x,s)=0$.

(iii) : לכל $x\in V$ מתקיים כי $c'(t,x)=0$.

הגדרה. <u>הרחבה של רשת זרימה</u> תהי $N=(V,E,c,s,t)$ רשת זרימה רגילה. נגדיר הרחבה של רשת זרימה זו, באופן הבא. ההרחבה היא $N'=(V,c',s,t)$ כאשר $c':V\times V\to\mathbb{R}_{\ge 0}$ המוגדרת באופן הבא: עבור $x,y\in V$ מתקיים כי $c'(x,y)=\begin{cases}c(x,y) & (x,y)\in E\\ 0 & (x,y)\notin E\end{cases}$.

הגדרה. <u>זרימה מורחבת ברשת זרימה מורחבת</u> תהי $N'=(V,c',s,t)$ רשת זרימה מורחבת. <u>זרימה מורחבת</u> ברשת זו היא $f':V\times V\to\mathbb{R}$ המקיימת את התכונות הבאות:

(i) : <u>אנטי סימטריה</u>: לכל $x,y\in V$ מתקיים כי $f'(x,y)=-f'(y,x)$.

(ii) : <u>אילוץ הקיבול</u>: לכל $x,y\in V$ מתקיים כי $f'(x,y)\le c'(x,y)$.

(iii) : <u>חוק שימור החומר</u>: לכל $x\in V\setminus\{s\}$ מתקיים כי $\sum_{v\in V} f'(x,v)=0$.

הגדרה. <u>הרחבה של זרימה</u> תהי $N=(V,E,c,s,t)$ רשת זרימה רגילה שבה אין שני קודקודים $x\neq y$ כך ש-$f(x,y)>0$ וגם $f(y,x)>0$. תהי $N'=(V,c',s,t)$ הרחבה של N. תהי f זרימה (רגילה) ברשת N, <u>ההרחבה</u> של f ברשת המורחבת N' מוגדרת באופן הבא: יהיו $x,y\in V$ נגדיר $f'(x,y)=\begin{cases}f(x,y) & (x,y)\in E\wedge f(x,y)>0\\ -f(y,x) & (y,x)\in E\wedge f(y,x)>0\\ 0 & \text{אחרת}\end{cases}$.

הגדרה. שטף של זרימה מורחבת f' מוגדר באופן הבא: $|f'|=\sum\limits_{x\in V} f\left(s,x\right)$.

למה. תהי N רשת זרימה רגילה ותהי f זרימה רגילה ברשת N. תהי N' ההרחבה של N ותהי f' ההרחבה של f לרשת N'. אזי
(i) : N' היא רשת זרימה מורחבת ו-f' זרימה מורחבת חוקית ברשת זו.
(ii) : מתקיים כי $|f'|=|f|$.

הגדרה. צמצום של רשת זרימה מורחבת תהי $N'=(V,c',s,t)$ רשת זרימה מורחבת. נגדיר $E=\{(x,y):c'(x,y)>0\}$ ונגדיר $c:E\to\mathbb{R}_{>0}$ באופן הבא, עבור $(x,y)\in E$ נגדיר $c(x,y)=c'(x,y)$, ונגדיר $N=(V,E,c,s,t)$ להיות ה-צמצום של הרשת.

הגדרה. צמצום של זרימה מורחבת תהי N' רשת זרימה מורחבת ותהי $N=(V,E,c,s,t)$ הצמצום של N'. תהי f' זרימה מורחבת ב-N', נגדיר $f:E\to\mathbb{R}_{\geq 0}$ באופן הבא. עבור $(x,y)\in E$ נגדיר $f(x,y)=\max\{f'(x,y),0\}$.

למה. תהי N' רשת זרימה מורחבת ותהי f' זרימה מורחבת ברשת זו. תהי N הצמצום של N' ו-f הצמצום של f'. אזי:
(i) : N היא רשת זרימה רגילה ו-f זרימה רגילה ברשת זו.
(ii) : $|f|=|f'|$.
(iii) : תהי N' הרחבה של רשת רגילה N, אזי הצמצום של N' מחזיר את הרשת N.

הערה. כל הגדרות אלה הן הבסיס לאלגוריתם פורד-פולקרסון, המרחיב את הרשת ומצמצם אותה חזרה. הרי גילינו לקורא כבר, כי מה שהאלגוריתם יעשה הוא בכל פעם למצוא מסלול $s\to t$, להזרים בו כמה שיותר, כאשר בכל פעם, תוזרם זרימה הפוכה בכיוון ההפוך, ותווצר רשת זרימה מורחבת עם זרימה מורחבת.

דוגמת הרצה - להוסיף

לקראת הכתיבה הפורמלית של האלגוריתם של $F\&F$ נרשום כמה הגדרות.

הגדרה. תהי $N=(V,c,s,t)$ רשת זרימה מורחבת ותהי f זרימה מורחבת ברשת זו. נגדיר:
(i) : הקיבול השיורי הוא פונקציה $c_f:V\times V\to\mathbb{R}_{\geq 0}$ המוגדרת באופן הבא. עבור $x,y\in V$ נגדיר $c_f(x,y)=c(x,y)-f(x,y)$.
(ii) : הרשת השיורית היא הרביעיה $N_f=(V,c_f,s,t)$.
(iii) : אוסף הצלעות השיורי הינו $E_f=\{(x,y)\mid c_f(x,y)>0\}$.
(iv) : הגרף השיורי הינו $G_f=(V,E_f)$.
(v) : מסילת הרחבה היא מסילה פשוטה בין s ל-t בגרף השיורי G_f.
(vi) : הקיבול השיורי של מסילת הרחבה p הוא $c_f(p)=\min\limits_{e\in p}\{c_f(e)\}$.
(vii) : הזרימה השיורית במסילת ההרחבה p היא פונקציה $\Delta_{f,p}:V\times V\to\mathbb{R}$ המוגדרת באופן הבא. $\Delta_{f,p}(x,y)=$

$$\begin{cases} c_f(p) & (x,y)\in p \\ -c_f(p) & (y,x)\in p \\ 0 & \text{אחרת} \end{cases}.$$

20.5 האלגוריתם של פורד ופולקרסון

קלט רשת זרימה רגילה $\tilde{N}$.

פלט זרימה אופטימלית g ברשת $\tilde{N}$.

אלגוריתם 22 האלגוריתם של פורד ופולקרסון למציאת זרימה אופטימלית ברשת זרימה

1. הרחבה נרחיב את הרשת $\tilde{N}$ לרשת מורחבת N.
2. אתחול נתאחל את f להיות זרימת האפס ברשת N.
3. איטרציה בכל שלב נמצא מסילת הרחבה p בגרף השיורי G_f ונעדכן $f = f + \Delta_{f,p}$.
4. עצירה כאשר אין מסילת הרחבה בגרף השיורי נעצור.
5. צמצום נצמצם את הרשת N חזרה לרשת הרגילה $\tilde{N}$ ואת הזרימה f כרגע ברשת לזרימה רגילה g ברשת $\tilde{N}$.
6. נחזיר את g.

יש לנו הרבה עבודה, עלינו להוכיח חוקיות ואופטימליות.

משפט. (לא פורמלי) האלגוריתם של $F\&F$ מחזיר זרימה חוקית ואופטימלית.

למה. תהי $N = (V, c, s, t)$ רשת זרימה מורחבת ותהי f זרימה ברשת זו. אזי:
(i) : הרשת השיורית N_f היא רשת זרימה מורחבת חוקית המאפסת לולאות, צלעות הנכנסות למקור והיוצאות מהבור.
(ii) : הזרימה השיורית $\Delta_{f,p}$ היא זרימה חוקית ברשת השיורית ו-$|\Delta_{f,p}| = c_f(p)$.
(iii) : הזרימה $f_1 = f + \Delta_{f,p}$ היא זרימה חוקית ברשת N וגם $|f_1| = |f| + c_f(p)$.

הוכחה: (הלמה)

(i) : מספיק לוודא כי עבור $x, y \in V$ מתקיים $c_f(x,y) \geq 0$. אמנם מכיון ש-f היא זרימה חוקית ברשת N, מתקיים $f(x,y) \leq c(x,y)$ ולכן $c_f(x,y) = c(x,y) - f(x,y) \geq 0$.

אנטי הסימטריה: האנטי סימטריה השיורית נובעת ישירות מההגדרה.

אילוץ הקיבול: יהיו $x, y \in V$ נבחין בין שני מקרים.

אם הצלע $\tilde{e} = (x,y) \in p$ אזי על פי הגדרת הזרימה השיורית $\Delta_{f,p}(\tilde{e}) = c_f(p) = \min_{e \in p}\{c_f(e)\} \leq c_f(\tilde{e})$. אחרת, אם $(x,y) \notin p$ אזי $\Delta_{f,p} \leq 0 \leq c_f(x,y)$.

חוק שימור החומר: עבור הזרימה השיורית, יהי $x \in V \setminus \{s,t\}$. אם x נמצא על המסילה, אזי $x \notin \{s,t\}$ ולכן קיים קודקוד y הקודם ל-x על המסילה וקודקוד z הבא אחרי x על המסילה. במקרה זה יוצאות מ-x בדיוק שתי צלעות עם זרימה שיורית שונה מאפס וצלעות אלה הן $(x,y), (x,z)$ ולכן

$$\sum_{v \in V} \Delta_{f,p}(x,v) = \Delta_{f,p}(x,y) + \Delta_{f,p}(x,z) = -c_f(p) + c_f(p) = 0$$

כרצוי. אחרת, אם x לא על המסילה, במקרה זה על כל הצלעות היוצאות ממנו הזרימה השיורית שווה לאפס ולכן כך גם הזרימה השיורית הכוללת היוצאת מ-x.

(ii) : השטף: של $\Delta_{f,p}$. יהי x הקודקוד על המסילה הבא אחרי קודקוד המקור s, אזי הצלע בין s ל-x זו הצלע היחידה היוצאת מקודקוד המקור שהזרימה השיורית בה שונה מאפס ולכן $|\Delta_{f,p}| = \sum_{v \in V} \Delta_{f,p}(s,v) = \Delta_{f,p}(s,x) = c_f(p)$.

(iii) : אנטי סימטריה של $f_1 = f + \Delta_{f,p}$. יהיו $x, y \in V$ אזי

$$f_1(y,x) = (f + \Delta_{f,p})(y,x) = f(y,x) + \Delta_{f,p}(y,x) = \underbrace{-f(x,y)}_{N \text{ זרימה חוקית ברשת}} \underbrace{-\Delta_{f,p}(x,y)}_{N_f \text{ זרימה חוקית ברשת}} = -f_1(x,y)$$

כרצוי.

<u>חוק שימור החומר ל-f_1</u>: יהי $x \in V \setminus \{s,t\}$ אזי

$$\begin{aligned}\sum_{v\in V} f_1(x,v) &= \sum_{v\in V}\left(f(x,v)+\Delta_{f,p}(x,v)\right)\\ &= \underbrace{\sum_{v\in V} f(x,v)+\sum_{v\in V}\Delta_{f,p}(x,v)}_{\text{זרימה חוקית ברשת}} = 0+0=0\end{aligned}$$

<u>אילוץ הקיבול</u>: יהיו $x,y \in V$ אזי

$$\begin{aligned}f_1(x,y) &= f(x,y)+\Delta_{f,p}(x,y) \le f(x,y)+c_f(x,y)\\ &= f(x,y)+\left(c(x,y)-f(x,y)\right) = c(x,y)\end{aligned}$$

<u>שטף</u>: נשאר לחשב את $|f_1|$. מתקיים

$$\begin{aligned}|f_1| &= \sum_{v\in V} f_1(s,v) = \sum_{v\in V}\left(f(s,v)+\Delta_{f,p}(s,v)\right) = \sum_{v\in V} f(s,v)+\sum_{v\in V}\Delta_{f,p}(s,v)\\ &= |f|+|\Delta_{f,p}| = |f|+c_f(p)\end{aligned}$$

כרצוי. ■

מסקנה. מהלמה נובע כי בכל שלב, לאחר כל איטרציה של האלגוריתם של $F\&F$ זורמת ברשת זרימה חוקית. כל איטרציה מעלה את שטף הזרימה ברשת שכן $c_f(p) > 0$.

הגדרה. <u>חתך</u> (S,T) ברשת זרימה $N=(V,c,s,t)$ הוא חלוקה זרה של קודקודי הרשת לשתי קבוצות S,T כך ש-$V = S[]\cup\cdot T$ עבורן $s \in S, t \in T$.

הגדרה. <u>הקיבול</u> של חתך (S,T) הוא $c(s,t) = \sum_{\substack{s\in S\\ y\in T}} c(x,y)$.

הגדרה. תהי f זרימה ברשת. <u>הזרימה</u> בחתך היא $f(S,T) = \sum_{\substack{x\in S\\ y\in T}} f(x,y)$.

טענה. תהי N רשת זרימה מורחבת ותהי f זרימה ברשת. יהי (S,T) חתך ברשת אזי $f(S,T) = |f|$.

$$f(y,x) > 0, f(x,y)$$

הוכחה: מתקיים

$$\begin{aligned}f(S,T) &= \sum_{\substack{x\in S\\ y\in T}} f(x,y) = \sum_{x\in S}\sum_{y\in T} f(x,y)\\ &= \sum_{x\in S}\left(\sum_{v\in V} f(x,v)-\sum_{u\in S} f(x,u)\right)\\ &= \sum_{x\in S}\sum_{v\in V} f(x,v)-\sum_{x\in S}\sum_{u\in S} f(x,u)\end{aligned}$$

מתקיים

$$\sum_{x\in S}\sum_{v\in V} f(x,v) = \sum_{v\in V} f(s,v) + \sum_{\substack{x\in S\\x\neq s}}\sum_{v\in V} f(x,v) = |f| + \underbrace{\sum_{\substack{x\in S\\x\neq s}} 0}_{\text{חוק שימור החומר}} = |f|$$

ואילו

$$\sum_{x\in S}\sum_{u\in S} f(x,u) = \sum_{\{x,u\}\subseteq S} \left(f(x,u) + f(u,x)\right)$$
$$\overset{\text{אנטי סימטריה}}{=} \sum_{\{x,u\}\subseteq S} 0 = 0$$

ולכן נובע כי $f(S,T) = |f|$. ■

מסקנה. יהי (S,T) חתך ברשת ותהי g זרימה ברשת. אזי $|g| \le c(S,T)$.

הוכחה: על פי הטענה שהוכחנו

$$|g| = g(S,T) = \sum_{\substack{x\in S\\y\in T}} g(x,y) \le \sum_{\substack{x\in S\\y\in T}} c(x,y) = c(S,T)$$

כרצוי. ■

משפט. (MinCut − MaxFlow) תהי N רשת זרימה ותהי f זרימה ברשת N. אזי שלושת התנאים הבאים שקולים זה לזה.
(i) : f זרימה אופטימלית ברשת N.
(ii) : אין מסילת הרחבה בגרף השיורי G_f.
(iii) : קיים חתך (S,T) כך ש-$c(S,T) = |f|$.

הוכחה: נוכיח $(i) \to (ii), (iii) \to (i), (ii) \to (iii)$.

$(ii) \to (i)$: נניח בשלילה שקיימת מסילת הרחבה p בגרף השיורי G_f. נגדיר $g = f + \Delta_{f,p}$, לפי מה שהוכחנו g היא זרימה חוקית ברשת N וכי $|g| = |f| + c_f(p) > |f|$ בסתירה לאופטימליות של f.

$(i) \to (iii)$: תהי g זרימה ברשת N. לפי המסקנה שהוכחנו $|g| \le c(S,T) = |f|$. על כן $|f|$ חסם מלעל על הזרימות ברשת ולכן היא אופטימלית.

$(iii) \to (ii)$: נגדיר $S = \{s\} \cup \left\{ \substack{x\in V \\ G_f \text{ ניתן להגיע מ-}s\text{ ל-}x\text{ בגרף}} \right\}$. נוודא כי $t \in T$ ולכן זה חתך חוקי ברשת. אמנם, על פי ההנחה אין מסילת הרחבה בגרף השיורי ולכן $s \in S, t \in T$.

יהי $x \in S$ ו-$y \in T$ נראה כי $f(x,y) = c(x,y)$. נניח בשלילה כי $f(x,y) \neq c(x,y)$, מכך לפי אילוץ הקיבול $f(x,y) < c(x,y)$ מה שגורר ש-$c(x,y) - f(x,y) = c_f(x,y) > 0$ לכן $(x,y) \in E_f$ כלומר (x,y) היא צלע בגרף השיורי G_f. נבנה מסלול מ-S ל-y בגרף השיורי G_f. נעבור על צלעות הגרף G_f מ-s ל-x (ניתן לעשות זאת כי $x \in S$) ונמשיך על הצלע (x,y). מכאן על פי הגדרת S נובע כי $y \in S$ וזו בסתירה לכך ש-$y \in T$. מכאן $f(x,y) = c(x,y)$ לכל $x \in S, y \in T$ ולכן

$$|f| = f(S,T) = \sum_{\substack{x\in S\\y\in T}} f(x,y) = \sum_{\substack{x\in S\\y\in T}} c(x,y) = c(S,T)$$

כרצוי. ■

מסקנות

1. אם האלגוריתם של פורד ופולקרסון עוצר, הוא מחזיר זרימה אופטימלית וזה כי התנאי לעצירתו הוא שאין מסילת הרחבה בגרף השיורי.
2. נניח כי האלגוריתם של פורד ופלקרסון עצר עם זרימה f ונריץ BFS על G_f מקודקוד המקור ברשת ועל ידי כך נגלה את כל הקודקודים שניתן להגיע אליהם מ-s, נסמן קבוצה זו ב-S אזי $(S, V \backslash S)$ הוא חתך מינימלי ברשת.

20.6 האלגוריתם של Edmonds Karp למציאת זרימה אופטימלית ברשת

קלט רשת זרימה רגילה $\tilde{N} = (V, E, \tilde{c}, s, t)$.

פלט זרימה אופטימלית g ברשת $\tilde{N}$.

אלגוריתם 23 האלגוריתם של אדמונדס וקארפ למציאת זרימה אופטימלית ברשת זרימה

1. הרחבה נרחיב את הרשת $\tilde{N}$ לרשת מורחבת N.
2. אתחול נתאחל את f להיות זרימת האפס ברשת N.
3. איטרציה בכל שלב נמצא מסילת הרחבה p בגרף השיורי G_f, באמצעות BFS, נבחר p להיות מסילת הרחבה בעלת אורך מינימלי (מספר צלעות מינימלי) ונעדכן $f = f + \Delta_{f,p}$.
4. עצירה כאשר אין מסילת הרחבה בגרף השיורי נעצור.
5. צמצום נצמצם את הרשת N חזרה לרשת הרגילה $\tilde{N}$ ואת הזרימה f ברשת לזרימה רגילה g ברשת $\tilde{N}$.
6. נחזיר את g.

משפט. האלגוריתם של אדמונדס וקארפ עוצר אחר $\mathcal{O}(|V| \cdot |E|)$ איטרציות.

מסקנה. זמן הריצה שלו הוא לכל היותר $\mathcal{O}\left(|V| \cdot |E|^2\right) = \mathcal{O}(|V| \cdot |E| \cdot (|E| + |V| + |V|))$ שכן $|V| = \mathcal{O}(|E|)$ כי הגרף קשיר.

נסתכל על $f_0, f_1, f_2, \ldots, f_N \equiv 0$ לכל זרימה f_i מתאים גרף שיורי G_{f_i} כאשר ב-G_{f_N} אין מסלול בין s ל-t. היינו רוצים לומר שבכל איטרציה של האלגוריתם, צלע אחת נעלמת בגרף, ואז לאחר מספר איטרציות מסוים לא יהיו יותר מסלולים בין s ל-t. נבחין כי אכן בכל שלב צלע נופלת, אבל מדוע הנימוק שלנו נופל? מכיוון שצלעות יכולות לחזור.

מסתבר, שבכל איטרציה המרחק של כל קודקוד מ-s גדל או נשאר אותו דבר. יתר על כן, עבור צלע (x, y) קריטית נגלה שהמרחק של x מ-s גדל כל פעם בלפחות 2, ולכן היא לא תוכל להופיע ביותר מ-$\frac{|V|}{2}$ גרפים. יש לנו $2|E|$ צלעות קריטיות אפשריות ולכן לאחר $2|E| \cdot \frac{|V|}{2} = |V||E|$ איטרציות כולן יתנתקו.

הגדרות

1. נסמן ב-f_i את הזרימה ברשת לאחר i איטרציות של האלגוריתם של אדמונדס-קארפ.
2. נסמן ב-P_i את מסילת ההרחבה שהאלגוריתם בוחר באיטרציה ה-i. אזי $f_i = f_{i-1} + \Delta_{f_{i-1}, P_i}$.
3. עבור $x \in V$ נסמן ב-$\delta_i(x)$ להיות המרחק של x מקודקוד המקור s בגרף השיורי G_{f_i}. כלומר $\delta_i(x)$ הוא האורך המינימלי של המסלול בין s ל-x ב-G_f אם אין מסלול כנ"ל נגדיר $\delta_i(x) = \infty$.

למה. (1) לכל $x \in V$ ולכל $i \geq 0$ מתקיים $\delta_{i+1}(x) \geq \delta_i(x)$ כלומר:
(i) : אם $\delta_i(x) < \infty$ אזי $\delta_{i+1}(x)$ הוא מספר סופי הגדול או שווה ל-$\delta_i(x)$ או $\delta_{i+1}(x) = \infty$.
(ii) : אם $\delta_i(x) = \infty$ אזי $\delta_{i+1}(x) = \infty$.

הערה. קצת אינטואיציה. נוכיח את שני המקרים בנפרד ואת שניהם נוכיח בשלילה. נתחיל במקרה (i). נניח כי קיימים x ו-i כך ש-$\infty > \delta_i(x) > \delta_{i+1}(x)$. מכל הקודקודים x עם התכונה הנ"ל נבחר את זה הקרוב ביותר ל-s בזמן $G_{f_{i+1}}$. נשים לב כי $x \neq s$. נביט בקודקוד y הנמצא לפני x במסלול הקצר ביותר המחבר את s ל-x בגרף $G_{f_{i+1}}$. אזי הצלע (y,x) לא יכולה להיות בגרף G_{f_i} שכן המרחק של y גדל ולכן המרחק של x גדל, וזה בא בסתירה להנחה שהמרחק שלו קטן יותר בגרף $G_{f_{i+1}}$.

מכאן (y,x) לא נמצאת בגרף G_{f_i} ולכן הצלע (y,x) רוויה בגרף G_{f_i} כלומר $f_i(y,x) = c(y,x)$ ואילו $f_{i+1}(y,x) < c(y,x)$ אבל מכאן הזרימה השיורית שהתווספה קטנה מאפס, ולכן (y,x) היא נגד כיוון המסילה, כלומר יש לנו מסילה קצרה ביותר P_i מ-s ל-t ש-(x,y) נמצאת עליה ומתקיים ש-$\delta_i(y) > \delta_i(x)$ שכן תת מסלול של מסלול קצר ביותר הוא קצר ביותר. אבל $\delta_{i+1}(y) > \delta_{i+1}(x)$ ולכן לא יתכן כי זמן לפני כן $\delta_i(x) < \delta_i(y)$.

הוכחה: נניח כי קיימים x ו-i כך ש-$\infty > \delta_i(x) > \delta_{i+1}(x)$. מכל הקודקודים x עם התכונה הנ"ל נבחר את זה הקרוב ביותר ל-s בזמן $G_{f_{i+1}}$.

נתבונן במסילה בעלת אורך מינימלי בין s ל-x בגרף $G_{f_{i+1}}$. יהי y הקודקוד הקודם ל-x על מסילה זו. מתקיים $\delta_{i+1}(y) = \delta_{i+1}(x) - 1 < \delta_{i+1}(x)$ ולכן y מקיים את טענת הלמה כלומר $\delta_i(y) \leq \delta_{i+1}(y)$ שכן אחרת נקבל סתירה למינימליות x. נסיק מכך כי $(y,x) \notin G_{f_i}$. אמנם, אם $(y,x) \in G_{f_i}$ היינו מקבלים

$$\delta_i(x) \leq \delta_i(y) + 1 \leq \delta_{i+1}(x) = \delta_{i+1}(x)$$

כלומר $\delta_i(x) \leq \delta_{i+1}(x)$ בסתירה לבחירה. לכן מתקיים $(y,x) \notin G_{f_i}$ כלומר $c_{f_i}(y,x) = 0$ כלומר $f_i(y,x) = c_{f_i}(y,x)$. מצד שני, $(y,x) \in G_{f_{i+1}}$ ולכן $f_{i+1}(y,x) < c(y,x)$ ולכן $f_{i+1}(y,x) < f_i(y,x)$. לפי דרך פעולתו של האלגוריתם זה שקול לכך ש-$\Delta_{f_i,P_i}(y,x) < 0$ שזה שקול לכך ש-$\Delta_{f_i,P_i}(x,y) > 0$ שזה שקול לכך ש-$(x,y) \in P_i$. מכיוון ש-P_i על פי הגדרתה היא מסלול באורך מינימלי בין s ל-t ב-G_{f_i} מתקיים $\delta_i(y) = \delta_i(x) + 1$. על כן

$$\delta_i(x) = \delta_i(y) - 1 \leq \delta_{i+1}(y) - 1 = \delta_{i+1}(x) - 2 < \delta_i(x) - 2$$

סתירה. נותר להוכיח את החלק השני.

עתה, נניח בשלילה כי קיים $x \in V$ ו-$i \geq 0$ כך ש-$\delta_i(x) = \infty$. מכל הקודוקדים עם התכונה הזו, נבחר קודקוד עם $\delta_{i+1}(x)$ מינימלי. נתבונן במסלול באורך מינימלי בין s ל-x ב-$G_{f_{i+1}}$. יהי y הקודקוד הקודם ל-x במסלול זה. נשים לב כי $x \neq s$ כי אחרת $\delta_i(x) = 0$ בסתירה להנחה.

נטען כי $\delta_i(y) < \infty$ ואמנם, $\delta_{i+1}(y) < \delta_{i+1}(x) < \infty$. אם $\delta_i(y) = \infty$ אז y גם היה סותר את טענת הלמה, אך מכיוון ש-$\delta_{i+1}(y) < \delta_{i+1}(x)$ זה היה סותר את בחירת x.

אם כן, $\delta_i(y) < \infty$. נטען כי הצלע (y,x) אינה צלע ב-G_{f_i}. אמנם, אם (y,x) הייתה צלע ב-G_{f_i}, ניתן היה להגיע מ-s ל-x ב-G_{f_i} באופן הבא: נגיע מ-s ל-y ונמשיך ל-x על גבי הצלע (y,x). לכן אנחנו בסיטואציה מוכרת מהמקרה הראשון: הצלע (y,x) לא שייכת ל-G_{f_i} אבל כן שייכת ל-$G_{f_{i+1}}$. ראינו בהוכחת המקרה הראשון כי זה גורר שהצלע (x,y) שייכת למסילת ההרחבה P_i אבל זה אומר ש-x נמצא על P_i. נזכור כי P_i היא מסילה מ-s ל-t בגרף השיורי G_{f_i}. אם x נמצא על P_i ניתן להגיע מ-s ל-x ב-G_{f_i} בסתירה לבחירת x. ■

הגדרה. הצלע (x,y) תקרא <u>קריטית</u> באיטרציה $i+1$ אם היא שייכת למסילת ההרחבה P_i ואם הקיבול השיורי שלה הוא המינימלי בין כל הצלעות של P_i, כלומר $c_{f_i}(x,y) = c_{f_i}(P_i)$.

למה. (2) נקבע $(x,y) \in V$. יהיו $i < k$ ונניח כי הצלע (x,y) היא צלע קריטית באיטרציות $i+1$ ב-$k+1$ של האלגוריתם. אזי $\delta_k(x) \geq \delta_i(x) + 2$.

הוכחה: נשים לב לכך שאם (x,y) הינה קריטית באיטרציה $i+1$ אזי (x,y) לא שייכת ל-$G_{f_{i+1}}$. כדי לראות זאת, מספיק לוודא כי $c_{f_{i+1}}(x,y)=0$ ואמנם

$$\begin{aligned} f_{i+1}(x,y) &= f_i(x,y)+\Delta_{f_i,P_i}(x,y) \overset{\text{קריטית}(x,y)}{=} f_i(x,y)+c_{f_i}(x,y) \\ &= f_i(x,y)+c(x,y)-f_i(x,y)=c(x,y) \end{aligned}$$

ולכן

$$c_{f_{i+1}}(x,y)=c(x,y)-f_{i+1}(x,y)=0$$

אם כן (x,y) אינה צלע ב-$G_{f_{i+1}}$. מצד שני מכיוון שהיא צלע קריטית באיטרציה $k+1$ היא שייכת ל-P_k ולכן היא צלע ב-G_{f_k}. כפי שראינו בהוכחת למה 1 זה גורר כי הזרימה ב-(x,y) חייבת לקטון באיטרציה $y+1$ עבור $i+1\leq y\leq k-1$ כלשהו. וזה קורה אם ורק אם הצלע בכיוון ההפוך שייכת למסילת ההרחבה הנבחרת באיטרציה זו, כלומר $(y,x)\in P_y$. מכאן

$$\begin{aligned} \delta_k(x) &\overset{\text{למה }1}{\geq} \delta_y(x) \overset{(y,x)\in P_y}{\geq} \delta_y(y)+1 \\ &\overset{\text{למה }1}{\geq} \delta_i(y)+1 \overset{(x,y)\in P_i}{\geq} \delta_i(x)+2 \end{aligned}$$

כרצוי. ■

למה. (עזר) אם (x,y) היא צלע קריטית איטרציה כלשהי של האלגוריתם אזי $(x,y)\in E$ או $(y,x)\in E$ כאשר E הוא אוסף הצלעות של הרשת המקורית $\tilde{N}=(V,E,c,s,t)$.

הוכחה: יהי $i\geq 0$ כך ש-(x,y) הינה קריטית באיטרציה $i+1$ של האלגוריתם. אזי (x,y) היא צלע ב-G_{f_i} ולכן

$$c_{f_i}(x,y)=c(x,y)-f_i(x,y)>0$$

נבחין כי יש שתי אפשרויות:

1. $c(x,y)>0$. מהגדרת הקיבול המורחב, זה אומר ש-$(x,y)\in E$.
2. $c(x,y)=0$ ולכן $f_i(x,y)<0$. מכאן, בגלל האנטי סימטריה של f_i מתקיים $f_i(y,x)>0$ ולכן מאילוץ הקיבול $c(y,x)\geq f_i(y,x)>0$ ולכן $(y,x)\in E$.

■

הוכחה: (אדמונדס וקארפ עוצר) נטען כי לכל $x,y\in V$ הצלע (x,y) יכולה להיות צלע קריטית בלכל היותר $\frac{|V|+1}{2}$ איטרציות. אמנם, נסמן ב-t מספר האיטרציות בהן (x,y) היא צלע קריטית. יהיו אלה האיטרציות $i_1+1,i_2+1,\ldots,i_t+1$ אזי לפי למה 2 לכל $1\leq j<y\leq t-1$ מתקיים

$$\delta_{i_y+1}(x)\geq\delta_{i_j}(x)+2$$

מכיוון ש-$\delta_{i_1}(x)\geq 0$ נקבל כי

$$\delta_{i_t}(x)\geq 2(t-1)$$

מצד שני, מכיוון שב-$G_{f_{i_t}}$ יש $|V|$ קודקודים, מתקיים $\delta_{i_t}(x)\leq|V|-1$ קיבלנו אם כך כי

$$2t-2\leq|V|-1$$

ולכן

$$t \leq \frac{|V|+1}{2}$$

עתה, נשים לב כי מלמת העזר נובע כי לכל היותר $2\,|E|$ צלעות יכולות להיות קריטיות. מכאן, מכיוון שבכל איטרציה יש צלע קריטית, מעקרון שובך היונים, מספר האיטרציות של האלגוריתם הוא לכל היותר

$$\frac{|V|+1}{2} \cdot 2\,|E| = O\left(|V| \cdot |E|\right)$$

כרצוי. ■

21 תרגול 9 - רשתות זרימה

נניח כי אנו מסוגלים לפתור בעיות רשתות זרימה, נציג בעיות נתונות בצורה זו.

21.1 זרימה ברשתות

הגדרה. רשת זרימה היא חמישיה (V,E,c,s,t) כאשר $G=(V,E)$ גרף מכוון. $s,t \in V$ על s נחשוב כמקור ו-t כבור. $c: E \to \mathbb{R}_{\geq 0}$ פונקציית קיבול.

הערה. רשתות אינטרנט נראות בדיוק ככה - כל ערוץ מסוג להעביר מידע בקיבולת מסוימת.

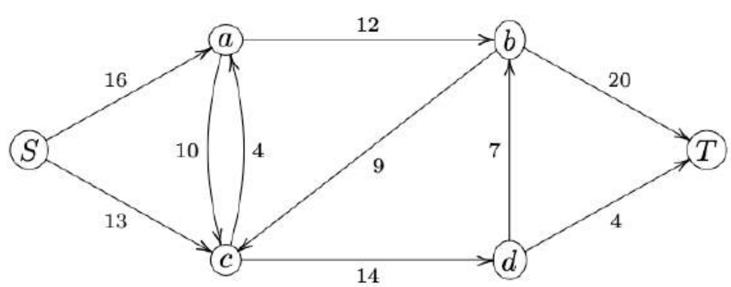
הגדרה. זרימה חוקית היא $f: E \to \mathbb{R}_{\geq 0}$ המקיימת את התכונות הבאות:

(i) : אילוץ הקיבול: לכל $e \in E$ מתקיים כי $f(e) \leq c(e)$.

(ii) : שימור החומר: לכל $v \in V \setminus \{s,t\}$ מתקיים כי $\sum_{e=(v,u)\in E} f(e) = \sum_{e=(u,v)\in E} f(e)$.

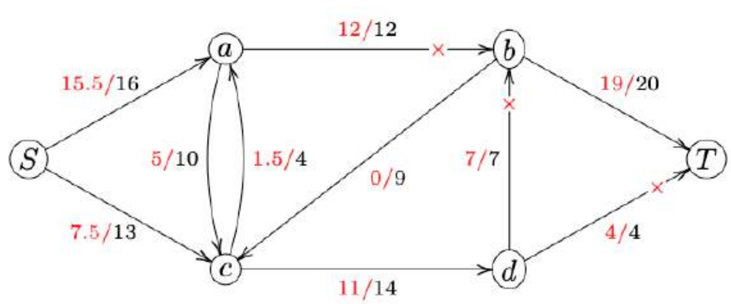
(iii) : שטף $\sum_{e=(s,v)\in E} f(e) = |f|$.

נביט בדוגמא הבאה:



איור 11: רשת זרימה

לרשת זו נוכל להגדיר את הזרימה הבאה:



איור 12: דוגמא לזרימה ברשת

נבחין כי כל הקודקודים משני הצדדים של d, b יוצרים חתך, שכן הצלעות $(a, b), (d, b)$ ברוויה. דבר זה יתרום לאינטואיציה שלנו בהמשך - קל יותר לחשוב על זרימה ברשת כחתך, ולא כזרימה.

21.2 רדוקציה לבעיות זרימה

נרצה לפתור בעיה A בעזרת פתרון לבעיה B. כלומר ניקח קלט ל-A, נמיר אותו לקלט ל-B, נריץ אלגוריתם עבור B ונמיר את הפלט חזרה לפלט של A.

הטענה הבאה היא טענה פילוסופית על הרדוקציה.

בהוכחת נכונות של אלגוריתם שמבצע רדוקציה נבצע את השלבים הבאים:

(i) : כל פתרון חוקי לבעיה המקורית ניתן להמרה לפתרון חוקי בבעיית "העד" (B) (אנחנו לא מפספסים פתרונות).

(ii) : ההמרה חזרה לפלט של הבעיה המקורית "משמרת חוקיות".

(iii) : ההמרות "משמרות ערך".

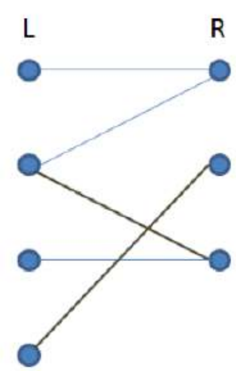
21.3 מציאת התאמה מקסימלית בגרף דו צדדי

קלט גרף דו צדדי (L, R, E) כלומר $E \subseteq L \times R$.

פלט שידוד מקסימלי בגרף $M \subseteq E$ עם $|M|$ מקסימלי, ובגרף (L, R, M) לכל קודקוד יש דרגה ≥ 1.

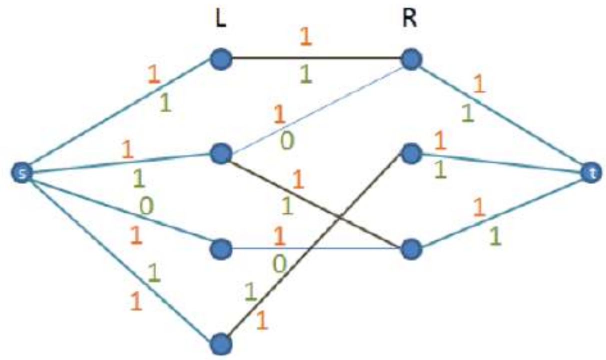
הערה. כשפתרנו בעיה באמצעות מטרואיד בעבר, התעלמנו מכך שבניית I היא אקספוננציאלית. עתה נראה אלגוריתם שבאמת רץ בזמן לינארי.

עולה השאלה, בהנתן הגרף



איור 13: גרף דו צדדי - כיצד נייצגו כרשת זרימה?

כדי לייצגו כרשת זרימה נוסיף קודקוד מקור וקודקוד בור. עולה השאלה איך נגדיר קיבול של כל קודקוד? אלגוריתם אדמונס קארפ מחזיר פתרון בשלמים בהנתן שכל הגרף בשלמים, לכן נדע שאם יש זרימה בצלע מקודקוד מסוים היא זרימה עם ערך שלם.



איור 14: דוגמא לקיבול ברשת הזרימה הנוצרת

נרשום אלגוריתם ליצירת רשת זרימה זו:

אלגוריתם 24 מציאת זיווג מושלם מקסימלי באמצעות רשת זרימה

1. נגדיר $V = L \cup R \cup \{s,t\}$.
2. נגדיר $\vec{E} = \{(l,r) \mid \{l,r\} \in E\}$, $E_L = \{(s,l) \mid l \in L\}, E_R = \{(r,t) \mid r \in R\}$.
3. נגדיר $c \equiv 1$ כלומר לכל e מתקיים כי $c(e) = 1$.
4. נריץ את EK על הרשת $N = \left(V, \vec{E} \cup E_L \cup E_R, c, s, t\right)$ ונקבל f.
5. נמיר חזרה לפלט לבעיה המקורית. נחזיר את $M = \{\{l,r\} \in E \mid l \in L, r \in R, f(l,r) = 1\}$.

הוכחת נכונות

טענה. פלט האלגוריתם הוא שידוך חוקי.

הוכחה: ראשית, מההגדרה, $M \subseteq E$. נראה כי מדובר בשידוך. נתבונן ב-$l \in L$ ונניח בשלילה ש-$\deg_{(L,R,M)} l > 1$. אזי קיימת יותר מצלע אחת $e \in E$ כך ש-$e \in M$ ומבניית M, קיימת יותר מצלע אחת $e \in \vec{E}$ כך ש-$f(e) = 1$, דבר זה נובע מכך ש-EK מחזיר פתרון בשלמים ומאילוץ קיבול. אבל, הקיבול הנכנס ל-l הוא 1 בסתירה לאילוץ הקיבל והשימור. ■

טענה. פלט האלגוריתם הוא אופטימלי.

הוכחה: נראה שהפתרון אופטימלי באופן הבא.

1. $|f| = |M|$.
2. בהנתן שידוך M' ניתן להמיר אותו לזרימה חוקית f' כך ש-$|f'| = |M|$.
3. נסיק כי M אופטימלי: נניח בשלילה שקיים שידוך חוקי M' כך ש-$|M'| > |M|$. אזי $|f'| = |M'| > |M| = |f|$ בסתירה לאופטימליות f, מנכונות EK.

נותר להוכיח את $1, 2$. נתחיל מ-1.

מתקיים כי

$$|f| = \sum_{l \in L} f(s,l) \overset{\text{חוק שימור החומר ומבנה הגרף}}{=} \sum_{e \in \vec{E}} f(e) \overset{\text{אילוץ הקיבול } f(e)\in[0,1] \text{ ומ } EK:\, f(e)\in\{0,1\}}{=} \sum_{\substack{e \in \vec{E} \\ f(e)=1}} 1 \overset{\text{הגדרת } M}{=} |M|$$

נוכיח את 2.

יהי שידוך M' נבנה f' חוקי כך ש-$|M'| = |f'|$. נגדיר זרימה על ידי כלילת כל הצלעות ב-M'.

$$f'(e) = \begin{cases} 1 & e = (l,r), l \in L, r \in R, \{l,r\} \in M' \\ 0 & e = (l,r), l \in L, r \in R, \{l,r\} \notin M' \\ 1 & (s,l) \in E_L \ \exists \{l,r\} \in M' \\ 1 & (r,t) \in E_R \ \exists \{l,r\} \in M' \\ 0 & \text{אחרת} \end{cases}$$

נראה כי זו זרימה חוקית.

מבנייה $f' : E \to \mathbb{R}_{\geq 0}$.

<u>אילוץ הקיבול</u>: מבנייה, $c \equiv 1$ ו-f' מזרימה רק 0 או 1.

<u>שימור החומר</u>: נתבונן ב-$l \in L$ נרצה להראות ש-$\sum_{(u,l) \in E'} f'(l,u) = \sum_{(u,l) \in E'} f'(u,l)$ או במילים אחרות $f'(s,l) = \sum_{(l,r) \in \vec{E}} f'(l,r)$.

יש לנו שני מקרים - l משתתף בזיווג או לא. אם l משתתף בזיווג כלומר קיים $\{l,r\} \in M'$ אזי $f'(s,l) = 1$ וקיים בדיוק קודקוד אחד r מחוקיות הזיווג עבורו $f'(l,r) = 1$ בכל $r \neq r'$ מתקיים $f'(l,r') = 0$. אחרת, אם l לא משתתף בזיווג אזי $f'(s,l) = 0$ ולכל $r \in R$ מתקיים כי $f'(l,r) = 0$.

נרצה להראות ש- $|M'| = |f'|$. מתקיים כי

$$|M'| = \sum_{(l,r) \in M'} 1 = \sum_{(l,r) \in \vec{E} \ s.t \ \{l,r\} \in M'} f'(l,r) = \sum_{(s,l) \in E_L} f'(s,l) = |f'|$$

כרצוי. ■ בהוכחה עקבנו בדיוק אחר הסכימה שרשמנו.

22 תרגול 10 - חתכים וחתך מינימלי

22.1 חתכים

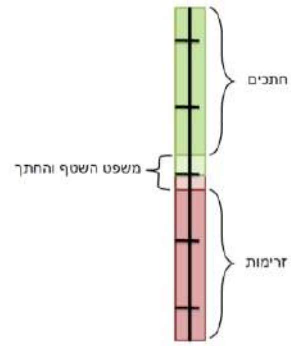
הגדרה. <u>חתך ברשתות זרימה</u> (S-T Cut) הוא חלוקה זרה של הקודקודים $V = S[]\cup \cdot T$ כלומר $S \cap T = \emptyset$ כך ש-$s \in S$ ו-$t \in T$.

הגדרה. <u>קיבול של חתך</u> מוגדר להיות $c(S,T) = \sum_{(u,v) \in S \times T} c(u,v)$.

טענה. לכל חתך (S,T) וזרימה חוקית f מתקיים $|f| \leq c(S,T)$.

משפט. (השטף והחתך) קיימת זרימה f וחתך S,T כך ש-$|f| = c(S,T)$.

מסקנה. קבוצת הפתרונות לבעיית ה-$MinCut$ חוסמים מלמעלה את קבוצת הפתרונות לבעיית ה-$MaxFlow$. יתר על כן, מהמשפט נובע כי להן נקודת מפגש:



איור 15: המחשה למשפט השטף והחתך

טענה. נניח שמצאנו f (S,T) כך ש-$|f|=c(S,T)$ אזי f בעלת שטף מקסימלי.

הוכחה: תהי f' זרימה חוקית אזי $|f'|\le c(S,T)=|f|$. ■

הערה. חתכים נוחים יותר להבנה ויזואלית מאשר זרימה מקסימלית.

22.2 שימושים של חתך מינימלי ברשת זרימה

נניח שהרצנו EK על השרת וקיבלנו זרימה מקסימלית. כיצד נמצא חתך מינימלי בגרף? נמחק את כל הצעות הרבויות ונריץ BFS או DFS דרך s. הצלעות שנגיע אליהן יהיו בהכרח צלעות בחתך של s. וריאציה על רעיון זה תאפשר לנו להוכיח באמצעות משפט השטף והחתך שנקבל שמדובר בחתך מינימלי.

22.2.1 בעיית המשקיעים והשחקנים

קלט $A=\{a_1,\dots,a_n\}$ קבוצת השחקנים ולכל i דרישת שכר s_i. $B=\{b_1,\dots,b_k\}$ קבוצת המשקיעים ולכל משקיע קבוצת שחקנים אהובה $A_i\subseteq A$ וכמות כסף d_i.

פלט $A'\subseteq A, B'\subseteq B$.

חוקיות לכל $b_i\in B'$ מתקיים כי $A_i\subseteq A'$.

אופטימליות נגדיר את הרווח של בחירה A',B' על ידי $p(A',B')=\sum\limits_{b_i\in B'}d_i-\sum\limits_{a_i\in A'}s_i$ נרצה לקבל $p(A',B')$ מקסימלי.

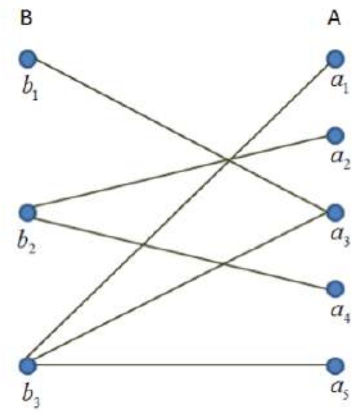
דוגמה. $k=3, n=5$ עם דרישות השכר הבאות:

s_1	s_2	s_3	s_4	s_5
70	150	200	80	100

ומידע על המשקיעים

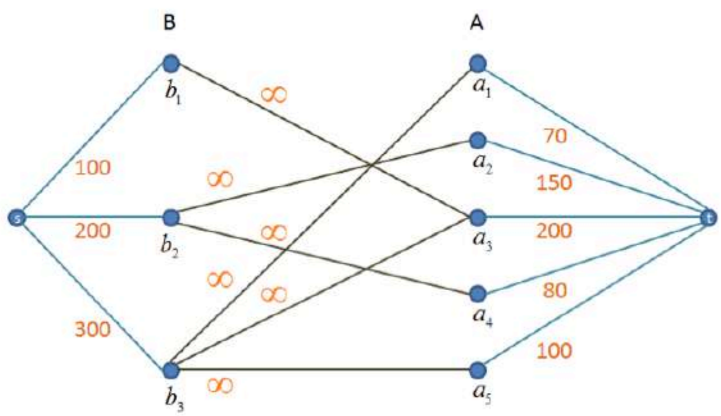
1	$\{a_3\}$	100
2	$\{a_2,a_4\}$	200
3	$\{a_1,a_2,a_3\}$	300

הדרך הנוחה להסתכל על כל המידע הזה היא באמצעות גרף עם צלעות מהמשקיעים לשחקנים שהם רוצים:



איור 16: הגרף בדוגמא

את גרף זה נהפוך לרשת הזרימה הבאה:



איור 17: רשת הזרימה המתאימה לגרף הקודם

נבחין כי מהגדרת פונקציית הערך, נרצה לקחת כמה שיותר משקיעים וכמה שפחות שחקנים. נרצה לחשוב על הבעיה כבעיית חתכים. נגדיר צלע בחתך להיות צלע מקודקוד שבחרנו לקודקוד שלא בחרנו. כלומר צלע שתצא מ-s לקודקוד b_i שלא בחרנו, תחשב צלע בחתך, ולכן נרצה למזער צלעות אלה ולקבל כמה שיותר קודקודים מ-B. צלעות היוצאות מ-a_i ל-t יחשבו גם בחתך וגם אותן נרצה למזער כי רוצים כמה שפחות שחקנים. יחד עם זאת, עלינו לדרוש עוד משהו - עלינו לוודא שאם לקחנו משקיע, ניקח תמיד כמה שיותר מהשחקנים שלו. כיצד נשיג זאת? נניח שלקחנו משקיע מסוים, אזי נרצה שכל הצלעות שיצאו ממנו לא יחתכו את החתך, ולכן ניתן להן קיבול ∞ והן בחיים לא יהיו בחתך, על כן נוכל לבחור את כל הצלעות היוצאות ממנו.

נציע את האלגוריתם הבא:

אלגוריתם 25 בעיית המשקיעים והשחקנים

1. נבנה את רשת הזרימה $N=(V,E,c,s,t)$ באופן הבא:
   (א) $V=\{s,t\}\cup A\cup B$.
   (ב) $E=\{(b_i,a_j)\mid \forall b_i\in B, a_j\in B_i\}\cup\{(s,b_i)\mid b_i\in B\}\cup\{(a_j,t)\mid a_j\in A\}$.
   (ג) $c(u,v)=\begin{cases} d_i & u=s,b=b_i \\ s_i & u=a_i,v=t \\ \infty & u=b_i,v=a_j \end{cases}$.
2. נמצא חתך S,T ברשת N בעזרת EK.
3. נגדיר $A'=A\cap S$ ו-$B'=B\cap S$.
4. נחזיר A',B'

הערה. אין כאן כמעט אזכור לרשתות זרימה. האלגוריתם שפותר את הבעיה הוא בעצם קופסא שחורה.

טענה. בהנתן S,T נגדיר $A'=A\cap S, B'=B\cap S$. אזי $c(S,T)$ סופי אם"ם A',B' חוקיים.

הוכחה: מתקיים כי $c(S,T)$ סופי $\iff$ אין אף קשת אינסופית שחוצה את החתך $\iff$ לכל $b_i\in S$ מתקיים לכל $a_j\in S$: $a_j\in A_i$ $\iff$אם $b_i\in S$ אז $A_i\subseteq S$ $\iff$ אם $b_i\in B'$ אזי $A_i\subseteq A'$. ■

טענה. יהי (S,T) חתך מקיבול סופי ויהיו A',B' המוגדרות לעיל. אזי $c(S,T)=D-p(A',B')$ עבור קבוע כלשהו D.

הוכחה: מתקיים כי

$$c(S,T)=\sum_{(u,v)\in S\times T}c(u,v)=\underbrace{\sum_{b_i\in T}c(s,b_i)}_{b_i\to a_i\text{-החתך מקיבול סופי ולכן אין צלעות מ}}+\sum_{a_i\in S}c(a_i,t)$$
$$=\sum_{i=1}^{n}d_i-\sum_{b_i\in S}d_i+\sum_{a_i\in S}a_i=\sum_{i=1}^{n}d_i-\left(\sum_{b_i\in S}d_i-\sum_{a_i\in S}a_i\right)$$
$$=\sum_{i=1}^{n}d_i-\left(\sum_{b_i\in B'}d_i-\sum_{a_i\in A'}a_i\right)=\sum_{i=1}^{n}d_i-p(A',B')$$

נגדיר $D=\sum_{i=1}^{n}d_i$ ונקבל את הרצוי. ■

מסקנה. האלגוריתם מחזיר פתרון אופטימלי.

הוכחה: נוכיח חוקיות ואופטימליות.

חוקיות: יש חתך סופי, למשל $\{s\},V'\setminus\{s\}$ עם קיבול D. האלגוריתם מחזיר חתך מינימלי ולכן מחזיר חתך סופי, ולכן מהלמה A',B' חוקיות.

אופטימליות: נניח בשלילה שיש A'',B'' חוקיות כך ש-$p(A',B')<p(A'',B'')$ אזי

$$c(S',T')=D-p(A',B')>D-p(A'',B'')=c(S'',T'')$$

בסתירה למינימליות S',T'. כאשר S'',T'' מוגדרת באופן הבא: $S''=\{s\}\cup A''\cup B''$ ו-$T''=V''\setminus S''$. ■

זמן ריצה

1. בניית הרשת - $n \times k$ צלעות באמצע ו-$n+k$ בקצוות.
2. הרצת EK ב-$\mathcal{O}\left(\left(nk\right)^2\left(n+k\right)\right) = \mathcal{O}\left(\left|E\right|^2\left|V\right|\right)$. באופן כללי, לאחר רדוקציה, חייבים לרשום זמן ריצה במונחי הקלט המקורי.
3. הגדרת הקבוצות - $n+k$ פעולות.

22.3 אלגוריתם FF

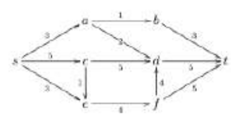
ניתן אינטואיציה לפעולת האלגוריתם.

הבעיה העיקרית שעלתה מהאלגוריתם הנאיבי שראינו בהרצאה היא שלא יכולנו "להתחרט" על הזרמת זרם דרך מסלול מסוים מ-s ל-t.

EK מאפשר להתחרט על ידי הזרמת זרימה שלילית והסתכלות על הגרף השיורי G_f. אם צריך להתחרט, האלגוריתם יבחר במסלול שיכיל צלעות שנוצרו עקב הזרמת זרם שלילי והפכו לזרם חיובי בגרף השיורי ונקבל "תיקון" לטעות.

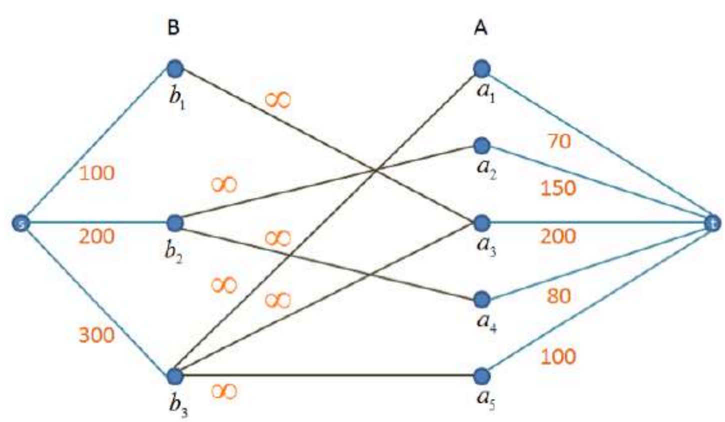
בהרצאה נוכיח את נכונותו.

לשם המחשה, ניתן דוגמת הרצה לאלגוריתם:

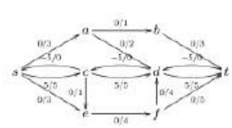


איור 18: רשת הזרימה המקורית

נתחיל עם זרימה $f(e) = 0$ נריץ את האלגוריתם EK עליה. נתחיל עם המסילה $s \to c \to d \to d$ נוכל להזרים דרכה 5 יחידות, ונקבל זרימה מורחבת:

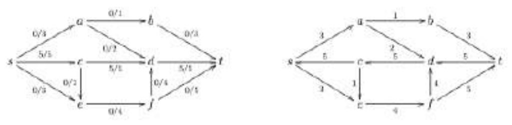


איור 19: רשת הזרימה המתאימה לגרף הקודם



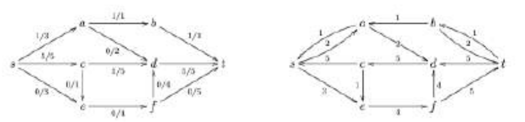
איור 20: הזרימה המרוחבת המתקבלת

לאחר מכן נוסיף את הזרימה לרשת המקורית ונבנה רשת שיורית על ידי חיסור ערכי הזרימה מהקיבול.



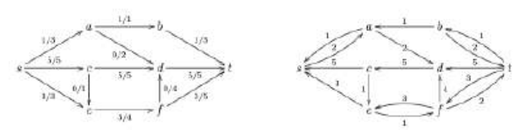
איור 21: מימין הרשת השיורית והרשת ומשמאל הזרימה ברשת המקורית

נחפש מסילה נוספת ברשת **השיורית**, למשל $s \to a \to b \to t$, ונזרים דרכה כקיבול הצלע הקטנה ביותר - 1. האלגוריתם יוסף את הזרימה המורחבת לרשת השיורית ומשם ייצור רשת שיורית חדשה, אולם בכדי להקל על הציור אנו נעשה את השני השלבים יחד. כך נקבל את הזרימה **הכוללת** ברשת המקורית עם הרשת השיורית **החדשה**:



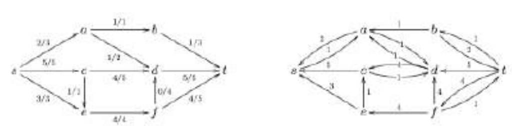
איור 22: מימין הרשת השיורית החדשה ומשמאל הזרימה הכוללת ברשת המקורית

עתה נחפש מסילה נוספת ברשת השיורית, $s \to e \to f \to t$ ונזרים דרכה כקיבול הצלע הקטנה ביותר - 3. נחשב כעת את הזרימה הכוללת ברשת המקורית עם הרשת השיורית החדשה



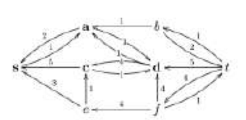
איור 23: מימין הרשת השיורית החדשה ומשמאל הזרימה הכוללת ברשת המקורית

נבחין כי קיימת מסילה נוספת ברשת השיורית $s \to a \to d \to c \to e \to f \to t$ ונזרים דרכה כקיבול הצלע הקטנה ביותר 1. נחשב כעת את הזרימה הכוללת ברשת המקורית עם הרשת השיורית החדשה.



איור 24: רשת הזרימהמימין הרשת השיורית החדשה ומשמאל הזרימה הכוללת ברשת המקורית המתאימה לגרף הקודם

נסתכל על הרשת השיורית האחרונה - אין בה אף מסילה מ-s אל t ולכן נחזיר את הזרימה כי שסכמנו על הרשת **המקורית**. איך נוודא שזוהי זרימה מקסימלית? עלינו למצוא חתך עם קיבול הזהה לזרימה. נבצע את הפעולה הבאה: נריץ DFS בגרף החלר מהקודקוד s ונמצא את כל הקודקודים שנוכל להגיע אליהם מ-s, שנסמנם באותיות בולטות.



איור 25: תוצאת הרצת DFS מ-s

קיבלנו רשימה $\{s, a, c, d\}$ ואלה יהיו קודקודי s ואילו $T = \{b, e, f, t\}$. אם נסתכל על חתך זה ברשת המקורית נראה שערכו 10, וזהו גם השטף של הזרימה שמצאנו. על כן מצאנו זרימה מקסימלית.

23 תרגול 11 - דואליות

בתכנון לינארי הצגנו בעייה

$$\begin{aligned} & \max_{x\in\mathbb{R}^n} c^T x \\ s.t \quad & Ax \le b \\ & x \ge 0 \end{aligned}$$

כאשר הקלט (c,b,A).

הגדרה. הבעיה הדואלית המתאימה לבעיה הנ"ל היא

$$\begin{aligned} & \min_{y\in\mathbb{R}^m} b^T y \\ s.t \quad & A^T y \ge c \\ & y \ge 0 \end{aligned}$$

23.1 דואליות חלשה

טענה. יהי $x\in\mathbb{R}^n$ חוקי עבור הבעיה הפרימאלית (המקורית) ויהי $y\in\mathbb{R}^m$ חוקי עבור הבעיה הדואלית, אזי $c^T x \le b^T y$.

הוכחה: מתקיים כי

$$c^T x \overset{x\ge 0}{\le} \left(A^T y\right)^T x = y^T A x = y^T (Ax) \overset{y\ge 0}{\le} y^T b$$

■

23.2 דואליות חזקה

משפט. (הדואליות החזקה) קיימים x,y חוקיים שעבורם $c^T x = b^T y$. לא נוכיח.

השערה. אם נציג את בעיית ה-$maxFlow$ כבעיית LP ונראה שבעיית ה-$minCut$ ניתנת להצגה כבעיה הדואלית המתאימה לה, נסיק את משפט שטף החתך.

הערה. ניתן להציג אותה בעיה גם בצורה דואלית וגם בצורה לינארית.

23.3 מציאת שטף מקסימלי

נראה כי בעיית מציאת זרימה עם שטף מקסימלי שייכת למשפחת בעיות התכנון הלינארי, בהנתן רשמת זרימה (V,E,c,s,t). נרצה למצוא שטף מקסימלי, כלומר

$$\begin{aligned} & \max_f |f| = \max_f \sum_{(s,\cdot)\in E} f(s,\cdot) \\ s.t \quad & \forall e: f(e) \le c(e) \\ & \forall v\in V\setminus\{s,t\} \qquad \sum_{(v,u)\in E} f(v,u) = \sum_{(u,v)\in E} f(u,v) \\ & \forall e: f(e)\ge 0 \end{aligned}$$

נהפוך את אילוץ שימור החומר לזוג אילוצים.

$$\forall v\in V\setminus\{s,t\}\qquad \sum_{(v,u)\in E} f(v,u)-\sum_{(u,v)\in E} f(u,v)\leq 0$$
$$\forall v\in V\setminus\{s,t\}\qquad \sum_{(v,u)\in E} f(v,u)-\sum_{(u,v)\in E} f(u,v)\geq 0$$

נסמן $m=|E|$. אם כך, נגדיר

$$\begin{aligned}&\max_{x\in\mathbb{R}^m} d^Tx\\ s.t\quad & Ax\leq b\\ & x\geq 0\end{aligned}$$

כאשר $d_{ij}=\mathbb{1}_{i=s}$ ו-$x=\begin{pmatrix}|\\ f_{ij}\\ |\end{pmatrix}\in\mathbb{R}^{|E|}$. אזי

$$b=\begin{pmatrix}|\\ c_{e_i}\\ |\\ -\\ 0\\ \vdots\\ 0\end{pmatrix}\in\mathbb{R}^{|E|+|V|-2}$$

ולכן

$$A=\begin{pmatrix}1&0&0&\dots&0\\0&1&0&\dots&0\\0&0&1&\dots&0\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\dots&1\\\hline &&M&&\\\hline &&-M&&\end{pmatrix}\in\mathbb{R}^{(|E|+2\cdot(|V|-2))\times|E|}$$

כאשר $M_{ve}=\begin{cases}1 & \exists u\in V,e=(u,v)\in E\\ -1 & \exists u\in V,e=(v,u)\in E\\ 0 & \text{אחרת}\end{cases}$ מייצגת את אילוץ שימור החומר, בכל שורה בה היא מייצגת את אילוץ שימור החומר לכל קודקוד. המטריצה $-M$ נותנת לנו את האילוץ השני של שימור החומר ככה שביחד הן נותנות את אילוץ השוויון.

כלומר, נוכל לחשוב על אילוץ שימור החומר באופן הבא:

$$\forall v\in V\setminus\{s,t\}\qquad \sum_{(v,u)\in E} f(v,u)+\sum_{(u,v)\in E}(-1)f(u,v)+\sum_{e\in E\ \text{לא נוגעת ב-}v} 0\cdot f(u,v)\leq 0$$
$$\forall v\in V\setminus\{s,t\}\qquad \sum_{(u,v)\in E} f(v,u)+\sum_{(v,u)\in E}(-1)\cdot f(u,v)+''''\leq 0$$

יחד עם זאת, אפשר לרשום את הבעיה בצורה אחרת על ידי הגדרת בעיה חדשה. זה נובע מכך שאנו יכולים לוותר על אילוץ שימור החומר המלא ולהשאר עם אילוץ החומר החלקי, ומכך לקבל זרימה חוקית על ידי המרה. ברור שלבעיה הנ"ל הפתרון האופטימלי נותן שטף גדול יותר מהפתרון האופטימלי של הבעיה המקורית, ולכן אם נראה שאפשר להמיר אותו לשמור על השטף, נוכל להסתפק בפתרון הבעיה החדשה והמרה. כלומר אנו מקבלים את הבעיה שהגדרנו עם

$$A=\begin{pmatrix}1&0&0&\dots&0\\0&1&0&\dots&0\\0&0&1&\dots&0\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\dots&1\\\hline &&M&&\end{pmatrix}\in\mathbb{R}^{(|E|+|V|-2)\times|E|}$$

טענה. תהי g זרימה חוקית בבעיה החדשה. נבנה f חוקית עבור המקורית המקיימת $|f|=|g|$.

הוכחה: אם יש מעגל עם זרימה חיובית ב-g, ניקח את הצלע עם הזרימה המינימלית על המעגל ונוריד את הזרימה הזו מכל צלעות המעגל. נחזור על התהליך עד שלא יהיו יותר מעגלים בגרף. כשבגרף אין מעגלים, נמיין את הקודקודים ב-$V\setminus\{s,t\}$ במיון טופולוגי ונקבל סדר $(v_1,\ldots,v_{n-2})$. נעבור על הקודקודים לפי הסדר ובכל v_i אם הזרימה היוצאת גדולה מהזרימה הנכנסת, נפחית זרימה מהצלעות היוצאות עד שנקבל שוויון.

לאורך כל התהליך הנ"ל לא נגענו ב-s ולכן נשארנו עם אותו שטף. נבחין כי ממיון הקודקודים, אנו לא צריכים להפחית לאחר שהפחתנו מהצלעות היוצאות. ■

23.4 הבעיה הדואלית

נרשום אותה במדויק לפי ההגדרה:

$$\begin{aligned} &\min_{y\in\mathbb{R}^{|E|+|V|-2}} b^T y \\ s.t \quad & A^T y \geq d \\ & y \geq 0 \end{aligned}$$

נראה שהיא מייצגת בעיית MinCut. נבחין כי אנו עושים מינימיזציה ל-$b^T y=\sum c(e_i)\, y_i$ וזה למעשה הקיבול של החתך. עתה

$$A^T y=\begin{pmatrix} 1 & 0 & 0 & \ldots & 0 & | & \\ 0 & 1 & 0 & \ldots & 0 & | & \\ 0 & 0 & 1 & \ldots & 0 & | & M^T \\ \vdots & \vdots & \vdots & \ddots & \vdots & | & \\ 0 & 0 & 0 & \ldots & 1 & | & \end{pmatrix}\begin{pmatrix} y_1 \\ \vdots \\ y_m \\ z_1 \\ \vdots \\ z_{n-2} \end{pmatrix}\geq\begin{pmatrix} | \\ \mathbb{1}_{i=s} \\ | \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

ולכן

$$A^T y \geq d$$

בבלוק הראשון, אנו מקבלים $1\cdot y_e$ בכל שורה לפי הצלע שאנו מסתכלים עליה.

עתה, עבור M^T נקבל שבכל עמודה יש -1 מצלע שיוצאת מ-v ו-1 מצלע שנכנסת. כלומר נקבל את האילוץ

$$\begin{aligned} &\forall e=(v,u)\,, v\neq s,t \quad 1\cdot y_e+1\cdot z_u-z_v\geq 0 \\ &\forall e=(s,u)\,, 1\cdot y_e+z_u\geq 1 \\ &\forall e=(v,t)\,, 1\cdot y_e-z_v\geq 0 \\ &(s,t) \quad 1\cdot y_e\geq 1 \end{aligned}$$

נתבונן בעמודה עבור צלע ספציפית. מופיע המקדם 1 בקוארדינטה המתאימה לאילוץ הקיבול. כמו כן, לכל צלע, כמעט, מופיע המקדם 1 בשורה המתאימה לאילוץ שימור החומר עבור הקודקוד אליו הצלע נכנסת ו--1 בשורה המתאימה לאילוץ שימור החומר עבור הקודקוד ממנו הצלע יוצאת. המקרים יוצאי הדופן הם הצלעות שיוצאות מ-s והצלעות שנכנסות אל t. עבורן לא מתקיים אילוץ שימור החומר בקודקודים הללו ולכן רק אחד המקדמים מופיע.

במילים אחרות $y_e=1$ אם הצלע e חותכת את החתך. מהמשוואה השנייה אם $y_e=0$ כלומר היא לא חותכת את החתך אז $z_u\geq 1$ ולכן נשער שאם $z_u\geq 1$ אז u בצד של s. אותו נימוק תופס גם למשוואה השנייה. במשוואה השלישית נקבל כי אם הצלע e לא חותכת את החתך אז הם באותו הצד.

מדואליות חזקה נסיק את משפט השטף והחתך.

חלק VI

התמרת פורייה $(\mathrm{DFT}\,\&\,\mathrm{FFT})$

24 מבוא

מרת פורייה בדידה מסומנת ב-DFT – Discrete Fourier Transform, האלגוריתם היעיל לחישוב שלה מסומן ב-FFT – Fast Fourier Transform.

בפיסיקה, מגדירים התמרת פורייה רציפה $\hat{f}\left(t\right)=\int\limits_{-\infty}^{\infty}f\left(x\right)e^{2\pi itx}\,\mathrm{dx}$ לכל $t\in\mathbb{R}$ עבור $f:\mathbb{R}\to\mathbb{R}$ שהאינטגרל עבורה מתכנס.

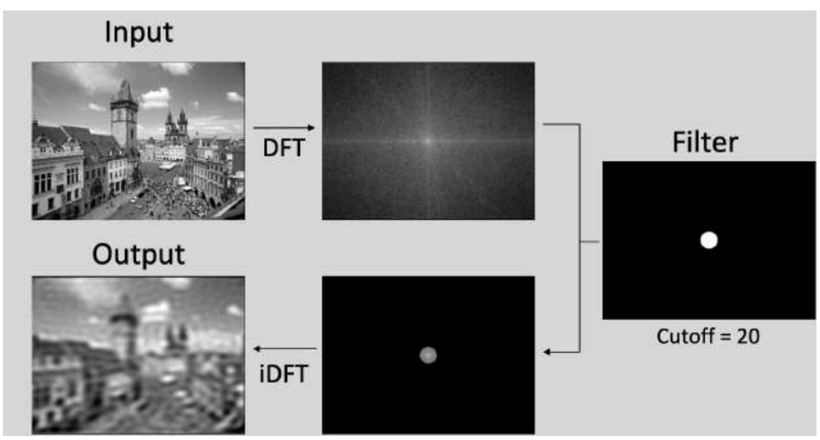
ניתן להגדירה עבור קבוצת פונקציות $f:[0,1]\to\mathbb{R}$ על ידי $\hat{f}\left(k\right)=\int\limits_{0}^{1}f\left(x\right)e^{2\pi ikx}\,\mathrm{dx}$ לכל $k\in\mathbb{Z}$. זה פירוק הרמוני.

את הצמצום לערכים טבעיים אפשר להפוך לסכום דיסקרטי ולהגדיר התמרת פורייה עבור $f:\mathbb{Z}_n\to\mathbb{R}$ על ידי $\hat{f}\left(k\right)=\frac{1}{n}\sum\limits_{x=0}^{n-1}f\left(x\right)e^{\frac{2\pi ikx}{n}}$ לכל $k\in\mathbb{Z}_n$.

מדוע הגדרות אלה שימושיות? ראשית, ניתן לשחזר את הפונקציה מההצגה בפוריה:

- $\mathbb{R}$ - $f\left(x\right)=\int\limits_{-\infty}^{\infty}\hat{f}\left(t\right)e^{-2\pi itx}\,\mathrm{dt}$.
- $\mathbb{T}$ - $f\left(x\right)=\int\limits_{0}^{1}\hat{f}\left(t\right)e^{-2\pi itx}\,\mathrm{dt}$, דבר זה הוא פירוק הפונקציה לתדרים למעשה, ובהמשך נראה למה זה שימושי.
- $\mathbb{Z}_n$ - $f\left(t\right)=\sum\limits_{x=0}^{n-1}\hat{f}\left(x\right)e^{-\frac{2\pi ixt}{n}}$.

למעשה, אנו מציגים את f לפי בסיס של אקפוננטים, ובכך מציגים אותה כסכום של "תדרים", העין האנושית למשל, קולטת יותר תדרים נמוכים, ולכן נוכל להשמיט מההצגה של f בפורייה תדרים גבוהים, ולראות איך זה נראה עבורנו לאחר שנפעיל טרנספורמציה הפוכה. למשל אם מדובר בתמונה, טבעי להפעיל פורייה, שכן מדובר בפונקציה דיסקרטית.



איור 26: כאן ניתן לראות הצגה של תמונה במרחב התדר (כפורייה) ולאחר מכן, הפעלת פילטר על הצגה זו, על ידי לקיחת התדרים הנמוכים בלבד. לאחר מכן, מופעלת הטרנספורמציה ההפוכה, ומתקבלת תמונה חדשה, שמכילה רק את התדרים הנמוכים של התמונה המקורית. תמונה זו דומה לתמונה המקורית רק שהיא מכילה פחות דיוק, וקווים כללים, או במילים אחרות, תדרים נמוכים בלבד. אותו דבר ניתן לבצע גם על תדרים גבוהים ועל כל קבוצת תדרים שנרצה. אפשר אף לעשות פילטר שממצע את התדרים, ולא רק מאפס, או כל טרנספורמציה אחרת.

25 פולינומים ופעולות על פולינומים

הגדרה. יהי $n\in\mathbb{N}$, נגדיר את V_{n-1} להיות מרחב הפולינומים ממעלה $\leq n-1$ מעל הממשיים.

נרצה לבצע בצורה יעילה את הפעולות הבאות על פולינומים:

- חיבור
- כפל
- הצבת ערך

עולה השאלה, כיצד נייצג פולינום.

הגדרה. <u>ייצוג המקדמים</u> של פולינום $p \in V_{n-1}$ $= \sum_{k=0}^{n-1} a_k x^k$ הוא וקטור המקדמים $\begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix}$.

טענה. (אלגברה לינארית) ההתאמה $p \to \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix}$ היא איזומורפיזם בין V_{n-1} ל-$\mathbb{R}^n$.

הערה. דרך אחרת להציג זאת היא שווקטור המקדמים הוא וקטור הקואורדינטות של p לפי הבסיס הסטנדרטי $\{1, x, x^2, \ldots, x^{n-1}\}$.

25.1 פעולות על פולינומים בייצוג המקדמים

חיבור

קלט שני פולינומים $P, Q \in V_{n-1}$ בייצוג המקדמים.

פלט הפולינום $R = P + Q \in V_{n-1}$ בייצוג המקדמים.

אלגוריתם 26 חיבור פולינומים בייצוג המקדמים

---

1. יהיו $\begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix}, \begin{pmatrix} b_0 \\ \vdots \\ b_{n-1} \end{pmatrix}$ וקטורי המקדמים של P, Q בהתאמה.
2. נחזיר את וקטור המקדמים של R שהוא $\begin{pmatrix} a_0+b_0 \\ \vdots \\ a_{n-1}+b_{n-1} \end{pmatrix}$
3. זמן הריצה הוא $\Theta(n)$.

הצבת ערך

קלט שני פולינומים $P \in V_{n-1}$ בייצוג המקדמים ומספר $x_0 \in \mathbb{R}$.

פלט $P(x_0)$

אלגוריתם 27 הצבת ערך בפולינום בייצוג המקדמים

---

1. כדי לחשב את $P(x_0) = \sum_{k=0}^{n-1} a_k x^k$ נחשב את סדרת החזקות של x_0 כלומר את $1, x_0, \ldots, x_0^{n-1}$ בזמן $\mathcal{O}(n)$, כאשר מחשבים את x_0^{k+1} על ידי $x_0 \cdot x_0^k$ לכל $0 \le k \le n-2$.
2. ואז נחשב את הסכום $P(x_0) = \sum_{k=0}^{n-1} a_k x^k$ בזמן $\mathcal{O}(n)$.
3. זמן ריצה $\Theta(n)$.

כפל

קלט שני פולינומים $P,Q\in V_{n-1}$ בייצוג המקדמים.

פלט הפולינום $R=P\cdot Q\in V_{2n-2}$ בייצוג המקדמים.

אלגוריתם 28 כפל פולינומים בייצוג המקדמים

1. נבחין איך נראים המקדמים של R. יהיו $\begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix},\begin{pmatrix} b_0 \\ \vdots \\ b_{n-1} \end{pmatrix}$ ווקטורי המקדמים של P,Q בהתאמה. אזי לכל $0\le m\le 2n-2$ מתקיים $\sum\limits_{i=0}^{m} a_i b_{m-i} = c_n=a_m b_0+a_{m-1}b_1+\ldots+a_0 b_m$ כאשר בנוסחא זו מניחים כי המקדמים a_k,b_k עבור $k\ge n$ שווים לאפס. באופן מפורש

$$\begin{aligned} c_0 &= a_0 b_0 \\ c_1 &= a_0 b_1 + a_1 b_0 \\ c_{n-1} &= a_0 b_{n-1} + \ldots + a_{n-1} b_0 \\ c_n &= a_1 b_{n-1} + \ldots + a_{n-1} b_1 \\ c_{2n-2} &= a_{n-1} b_{n-1} \end{aligned}$$

שזה סכום שהולך וגדל עד האיבר ה-$n-1$ וקטן החל ממנו עד האיבר ה-$2n-2$.

2. כל חישוב מקדם עולה $\mathcal{O}\left(n\right)$ ויש $\mathcal{O}\left(n\right)$ מקדמים ולכן זמן הריצה הוא $\Theta\left(n^2\right)$. יקר מדי.

25.2 פעולות על פולינומים בייצוג הערכים

רעיון נגדיר ייצוג אחר של פולינומים.

טענה. לפולינום ממעלה $\le n-1$ מעל שדה $\mathbb{F}$ כלשהו שהוא לא פולינום האפס, יש לכל היותר $n-1$ שורשים שונים בשדה.

מסקנה. הערכים של פולינומים ממעלה $\le n-1$ ב-n נקודות שונות קובעות את הפולינום בצורה יחידה.

הוכחה: (המסקנה) תהיינה $x_0,\ldots,x_{n-1}$ n נקודות שונות בשדה ונניח בשלילה כי קיימים שני פולינומים $P\neq Q$ ממעלה $\le n-1$ כך ש-$P\left(x_i\right)=Q\left(x_i\right)$ נגדיר $R=P-Q$. אזי R פולינום ממעלה $\le n-1$ כך ש-$R\left(x_i\right)=0$ לכל $0\le i\le n-1$ ולכן מהעובדה האלגברית, R הוא פולינום האפס ולכן $P=Q$ בסתירה להנחה. ■

הגדרה. תהיינה $x_0,x_1,\ldots,x_{n-1}$ n נקודות ממשיות שונות. ייצוג הערכים של הפולינום $P\in V_{n-1}$ הוא ווקטור הערכים של P הוא $\begin{pmatrix} P(x_0) \\ \vdots \\ P(x_{n-1}) \end{pmatrix}$.

שאלה לפי איזה בסיס הייצוג הנ"ל הוא ווקטור הקוארדינטות של P?

הגדרה. תהי $\begin{pmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \ldots & x_2^{n-1} \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^{n-1} \end{pmatrix} = A\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ מוגדרת להיות מטריצת ונדרמונדה.

משפט. $\det A=\prod\limits_{i<j}\left(x_i-x_j\right)$.

הוכחה: נוכיח באינדוקציה על n.

בסיס: $n=2$ מתקיים כי $A=\left(\begin{smallmatrix}1 & x_1\\ 1 & x_2\end{smallmatrix}\right)=x_2-x_1$.

שלב: נניח עבור $n-1$ נוכיח עבור n.

נפתח לפי השורה האחרונה

$$\det A=(-1)^{n+1}\cdot 1\cdot M_{n1}+\ldots+(-1)^{n+n}x_n^{n-1}M_{nn}$$

נבחין כי קיבלנו פולינום במשתנה x_n וכי אם $x_n=x_i$ עבור $1\le i\le n-1$ קיבלנו דטרמיננטה עם שתי שורות שוות ולכן

$$\det A=R\left(x_n-x_1\right)\left(x_n-x_2\right)\cdot\ldots\cdot\left(x_n-x_{n-1}\right)$$

נבחין כי R הוא המקדם של x_n^{n-1} ולכן

$$R=M_{nn}=\det\begin{pmatrix}1 & x_1 & x_1^2 & \ldots & x_1^{n-2}\\ 1 & x_2 & x_2^2 & \ldots & x_2^{n-2}\\ \vdots & \vdots & \ddots & & \vdots\\ \vdots & \vdots & & \ddots & \vdots\\ 1 & x_{n-1} & x_{n-1}^2 & \ldots & x_{n-1}^{n-2}\end{pmatrix}\overset{\text{הנחת האינדוקציה}}{=}\prod_{i<j}^{i,j\le n-1}\left(x_j-x_i\right)$$

ולכן

$$\begin{aligned}\det A&=\left(\prod_{i<j}^{i,j\le n-1}\left(x_j-x_i\right)\right)\cdot\left(x_n-x_1\right)\left(x_n-x_2\right)\cdot\ldots\cdot\left(x_n-x_{n-1}\right)\\ &=\prod_{i<j}\left(x_j-x_i\right)\end{aligned}$$

כרצוי. ■

הגדרה. נסמן ב-A_j את מטריצת ונדרמונדה שבה המשתנה ה-j הינו המשתנה החופשי x.

מסקנה. יהיו $x_1,\ldots,x_n$ n שונות בשדה ו-$y_1,\ldots,y_n$ נקודות בשדה $\mathbb{F}$, אזי הפולינום $P\left(x\right)=\sum_{i=1}^{n}\frac{\det A_i(x)}{\det A}y_i$ מקיים כי $P\left(x_i\right)=y_i$ לכל $1\le i\le n$ והוא היחיד ממעלה $n-1$ שמקיים זאת.

הוכחה: יהי $1\le i\le n$ אזי $\det A_i\left(x_i\right)=\prod_{k<l}\left(x_l-x_k\right)$ כאשר זו פונקציה המשתנה x_i. לכל $j\neq i$ נוכל להציב $x_i=x+j$ ומתקיים כי $\left(x_j-x_i\right)=\left(x_j-x_j\right)=0$ ולכן $\det A_i\left(x_j\right)=0$. עבור ההצגה x_i נקבל $\det A_i\left(x_i\right)=\det A$ ולכן

$$P\left(x_i\right)=\sum_{i=1}^{n}\frac{\det A_i\left(x_i\right)}{\det A}y_i=\frac{\det A_i\left(x_i\right)}{\det A}y_i=y_i$$

מהמסקנה הקודמת נובע כי הוא יחיד. אף על פי כן, ניתן להסיק יחידות גם מכך שאנו מחפשים פולינום $P=\sum_{i=0}^{n-1}a_ix^i$ כך ש-

$$P\left(x_j\right)=\sum_{i=0}^{n-1}a_ix_j^i=y_j$$

ניתן לרשום דרישה על ידי כך ש-

$$A\begin{pmatrix}a_0\\ \vdots\\ a_{n-1}\end{pmatrix}=\begin{pmatrix}y_0\\ \vdots\\ y_{n-1}\end{pmatrix}$$

מהיות $\det A \neq 0$ כי $x_0, \ldots, x_{n-1}$ שונים, מתקבל כי A הפיכה ולכן על ולכן קיימים $a_0, \ldots, a_{n-1}$ כנ"ל. אבל היא גם חח"ע ולכן המקדמים הנ"ל יחידים. ■

הערה. ניתן לרשום את המחוברים בצורה פשוטה יותר, והיא $\frac{\det A_m(x)}{\det A} = \frac{(x-x_0)\cdot(x-x_1)\cdot\ldots\cdot(x-x_{m-1})\cdot(x-x_{m+1})\cdots\cdot(x-x_{n-1})}{(x_m-x_0)(x_m-x_1)\cdot\ldots\cdot(x_m-x_{m-1})(x_m-x_{m+1})\cdot\ldots\cdot(x_m-x_{n-1})}$, במקום לחשב דטרמיננטות שלמות.

טענה. הסדרה $\left(\frac{\det A_1}{\det A}, \ldots, \frac{\det A_n}{\det A}\right)$ היא בסיס למרחב הפולינומים V_{n-1}.

הוכחה: מספיק להוכיח כי הסדרה בלתי תלויה לינארית, כי אז היא תהיה קבוצה בת"ל מקסימלית.

יהיו $\alpha_1, \ldots, \alpha_n$ עבורם $P = \sum_{i=1}^{n} \alpha_i \frac{\det A_i(x)}{\det A} = 0$ לכל $x \in \mathbb{R}$. מהטענה הקודמת, הפולינום P הוא הפולינום היחיד ממעלה $n-1$ שמקיים כי $P(x_i) = \alpha_i$ אבל מההנחה, $P = 0$ ולכן $\alpha_i = 0$ לכל $1 \leq i \leq n$. ■

מסקנה. עבור הבסיס $\mathcal{B} = \left(\frac{\det A_1}{\det A}, \ldots, \frac{\det A_n}{\det A}\right)$ מתקיים כי $[P]_{\mathcal{B}} = \begin{pmatrix} P(x_1) \\ \vdots \\ P(x_n) \end{pmatrix}$.

מסקנה. ההתאמה $P \to \begin{pmatrix} P(x_1) \\ \vdots \\ P(x_n) \end{pmatrix}$ היא גם איזומורפיזם בין V_{n-1} לבין $\mathbb{R}^n$.

חיבור

קלט שני פולינומים $P, Q \in V_{n-1}$ בייצוג הערכים.

פלט הפולינום $R = P + Q$ בייצוג הערכים.

אלגוריתם 29 חיבור פולינומים בייצוג המקדמים

---

1. מתקיים $\begin{pmatrix} R(x_0) \\ \vdots \\ R(x_{n-1}) \end{pmatrix} = \begin{pmatrix} (P+Q)(x_0) \\ \vdots \\ (P+Q)(x_{n-1}) \end{pmatrix} = \begin{pmatrix} P(x_0)+Q(x_0) \\ \vdots \\ P(x_{n-1})+Q(x_{n-1}) \end{pmatrix}$
2. זמן ריצה $\Theta(n)$.

כפל

קלט שני פולינומים $P, Q \in V_{n-1}$ בייצוג הערכים.

פלט הפולינום $R = P \cdot Q \in V_{2n-2}$ בייצוג הערכים.

אלגוריתם 30 כפל פולינומים בייצוג המקדמים

---

1. מתקיים $\begin{pmatrix} R(x_0) \\ \vdots \\ R(x_{n-1}) \end{pmatrix} = \begin{pmatrix} (P\cdot Q)(x_0) \\ \vdots \\ (P\cdot Q)(x_{n-1}) \end{pmatrix} = \begin{pmatrix} P(x_0)\cdot Q(x_0) \\ \vdots \\ P(x_{n-1})\cdot Q(x_{n-1}) \end{pmatrix}$
2. זמן ריצה $\Theta(n)$.

זה לא עובד, כי זה לא קובע את R באופן יחיד.

הפתרון הוא שנדרוש מראש את ייצוג הערכים של P ו-Q ב-$2n-1$ נקודות $x_0, \ldots, x_{2n-2}$ ואז כשנכפיל את P, Q נקבל $2n-1$ ערכים של R שיקבעו את R באופן יחיד. בהנחה שיש לנו נקודות אלה הפתרון הוא לינארי ב-n.

הצבת ערך

קלט פולינום $P \in V_{n-1}$ בייצוג הערכים ומספר $x_0 \in \mathbb{R}$.

פלט $P\left(x_0\right)$.

זו בעיית האינטרפולציה הפולינומית.

25.3 פתרון בעיית האינטרפולציה הפולינומית

נגדיר n פולינומים $L_0, \ldots, L_{n-1}$ ממעלה $n-1 \geq$ עם התכונה הבאה:

לכל $0 \leq k, m \leq n-1$ מתקיים $L_m\left(x_k\right)=\delta_{km}=\begin{cases}1 & k=m\\ 0 & k \neq m\end{cases}$. נגדיר אם כך

$$L_m(x)=\frac{\left(x-x_0\right)\left(x-x_1\right) \cdot \ldots \cdot\left(x-x_{m-1}\right) \cdot\left(x-x_{m+1}\right) \ldots \cdot\left(x-x_{n-1}\right)}{\left(x_m-x_0\right)\left(x_m-x_1\right) \cdot \ldots \cdot\left(x_m-x_{m-1}\right)\left(x_m-x_{m+1}\right) \cdot \ldots \cdot\left(x_m-x_{n-1}\right)}$$

טענה. יהי $P \in V_{n-1}$ אזי $P=\sum_{m=0}^{n-1} P\left(x_m\right) L_m$. בפרט $P\left(x_n\right)=\underbrace{\sum_{m=0}^{n-1} P\left(x_m\right) \underbrace{L_m\left(x_n\right)}_{\Theta(n)}}_{\Theta\left(n^2\right)}$.

מסקנה. ניתן לפתור את בעיית הצבת הערך בזמן $\Theta\left(n^2\right)$.

ראינו כי פעולת הכפל ירה בייצוג המקדמים וזולה בייצוג הערכים ופעולת הצבת הערך יקרה בייצוג הערכים וזולה בייצוג המקדמים.

25.4 מעבר בין הייצוגים

נתון ייצוג המקדמים $\begin{pmatrix} a_0 \\ \vdots \\ a_{n-1}\end{pmatrix}$ של פולינום $P \in V_{n-1}$ ו-n נקודות שונות $x_0, \ldots, x_{n-1}$. איך נחשב את ווקטור הערכים $\begin{pmatrix} P(x_0) \\ \vdots \\ P(x_{n-1})\end{pmatrix}$?
נבחין כי שני הווקטורים מציגים את P לפי שני בסיסים שונים, ולכן עלינו לכפול במטריצת מעבר.

למה. תהיינה $x_0, \ldots, x_{n-1}$ n נקודות ממשיות שונות ויהי $P \in V_{n-1}$ עם וקטור מקדמים $\begin{pmatrix} a_0 \\ \vdots \\ a_{n-1}\end{pmatrix}$ אזי $M\begin{pmatrix} a_0 \\ \vdots \\ a_{n-1}\end{pmatrix}=\begin{pmatrix} P(x_0) \\ \vdots \\ P(x_{n-1})\end{pmatrix}$

כאשר $M=\begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \ldots & x_1^{n-1} \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \ldots & x_{n-1}^{n-1}\end{pmatrix}$ היא מטריצת Van der Monde.

מסקנה. $\begin{pmatrix} a_0 \\ \vdots \\ a_{n-1}\end{pmatrix}=M^{-1}\begin{pmatrix} P(x_0) \\ \vdots \\ P(x_{n-1})\end{pmatrix}$.

הערה. זמן הריצה של כפל מטריצה בווקטור נתון הוא $\Theta\left(n^2\right)$ והפיכת מטריצה עולה $\Theta\left(n^3\right)$.

הוכחה: (הוכחת הלמה) יהי $0 \leq k \leq n-1$ אזי

$$\left(M\begin{pmatrix} a_0 \\ \vdots \\ a_{n-1}\end{pmatrix}\right)_k=\sum_{m=0}^{n-1} M(k, m) \cdot a_m=\sum_{m=0}^{n-1} x_k^m a_m=\sum_{m=0}^{n-1} a_m x_k^m=P\left(x_k\right)$$

כרצוי. ■

רעיון במקום לחשב את הערכים של פולינום בנקודות ממשיות, נחשב אותן ב-n נקודות מרוכבות שונות.

ההצדקה לזה היא ששדה הממשים $\mathbb{R}$ הוא תת שדה של שדה המרוכבים $\mathbb{C}$ ולכן ניתן להציב בפולינום $P \in V_{n-1}$ נקודות מרוכבות. העובדה האלגברית שלפולינום שונה מאפס ב-V_{n-1} יש לכל היותר $n-1$ שורשים מרוכבים שונים עדיין נכונה וכמסקנה מכך ניתן להציג כל פולינום כזה באופן יחיד עם הערכים שלו ב-n נקודות מרוכבות שונות.

26 התמרת פורייה בדידה DFT

26.1 מספרים מרוכבים

הגדרה. המספרים המרוכבים הם כל המספרים מהצורה $\mathbb{C} = \{a + bi \mid a, b \in \mathbb{R}\}$.

ניתן לרשום את המספר $a+bi$ במישור המרוכב, כנקודה $\binom{a}{b}$ במישור הקרטזי, רק שכאן ציר ה-y מייצג מספרים מדומים טהורים מהצורה bi וציר ה-x מייצג את הישר הממשי a.

נבחין כי כל מספר כנ"ל ניתן לרשום בצורה פולרית, עם θ, R כאשר $R = \sqrt{a^2+b^2}$ אשר $\mathbb{C} = \{R(\cos\theta + i\sin\theta) \mid \theta \in \mathbb{R}, R \geq 0\}$.

הכפל של מספרים מרוכבים בייצוג פולרי הינו

$$R_1(\cos\theta + i\sin\theta) \cdot R_2(\cos\varphi + i\sin\varphi) = R_1R_2(\cos(\theta+\varphi) + i\sin(\theta+\varphi))$$

מסמנים $R(\cos\theta + i\sin\theta) = Re^{\theta i}$ ואז הכפל מוצג בצורה אינטואיטיבית $R_1R_2e^{(\theta+\varphi)i}$.

26.1.1 מעגל היחידה המרוכב

הגדרה. מעגל היחידה המרוכב מוגדר להיות $\mathbb{T} = \{\cos\theta + i\sin\theta \mid 0 \leq \theta \leq 2\pi\}$.

מסקנה. $i = \cos\left(\frac{\pi}{2}\right) + i\sin\left(\frac{\pi}{2}\right)$ וגם $-i = \cos\left(\frac{3\pi}{2}\right) + i\sin\left(\frac{3\pi}{2}\right)$

מסקנה. אם z_1, z_2 על מעגל היחידה, גם z_1z_2 על מעגל היחידה.

מסקנה. יהי $z = \cos\theta + i\sin\theta$ ויהי $k \in \mathbb{Z}$ אזי $z^k = \cos k\theta + i\sin k\theta$.

הגדרה. שורשי היחידה מסדר n הם כל המספרים המרוכבים z המקיימים $z^n = 1$

הגדרה. נסמן $\omega_n = \cos\left(\frac{2\pi}{n}\right) + i\sin\left(\frac{2\pi}{n}\right) = e^{\frac{2\pi}{n}i}$.

נראה כי החזקות של ω_n הן כל שורשי היחידה השונים. עבור $0 \leq k \leq n-1$ מתקיים כי $\omega_n^k = \cos\left(\frac{2\pi k}{n}\right) + i\sin\left(\frac{2\pi k}{n}\right) = e^{\frac{2\pi k}{n}i}$.

משפט. שורשי היחידה מסדר n הם כל המספרים המרוכבים מהצורה ω_n^k כאשר $0 \leq k \leq n-1$.

הוכחה: מתקיים עבור $0 \leq k \leq n-1$ כי $\left(\omega_n^k\right)^n = \left(\omega_n^n\right)^k = 1^k = 1$ וקל לראות כי החזקות שונות אחת מהשנייה לכן קיבלנו n שורשי יחידה שונים ומכך שלפולינום $z^n - 1$ יש לכל היותר n שורשים שונים, אלה כל שורשי היחידה. ■

מסקנה. המספרים הנ"ל הם כל המספרים $R(\cos\theta + i\sin\theta)$ המקיימים $n\theta = 2\pi k$ וגם $R^n = 1$ כאשר R ממשי כלומר $R = 1$. במילים אחרות, שורשי היחידה מחלקים את המעגל ל-n חלקים שווים ולכן ניתן לרשום אותם כ-$e^{\frac{2\pi k}{n}i}$ שכן $e^{2\pi ki} = 1$. כאשר $0 \leq k \leq n-1$ שכן אחרת מקבלים כפילויות. הפלא ופלא קיבלנו בדיוק את בסיס פורייה. בנוסף מכך ש-$R = 1$ אלה מספרים שנמצאים על מעגל היחידה המרוכב.

הגדרה. התמרת פורייה בדידה מסדר n (DFT) יהי $\begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix} \in \mathbb{C}^n$ ויהי $P \in V_{n-1}$ $= \sum_{k=0}^{n-1} a_k z^k$. התמרת הפורייה הבדידה על P מוגדרת להיות

$$\mathrm{DFT}_n \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} P\left(\omega_n^0\right) \\ \vdots \\ P\left(\omega_n^{n-1}\right) \end{pmatrix}$$

מסקנה. התמרת פורייה DFT_n הינה מעבר מייצוג מקדמים של פולינום לייצוג הערכים שלו בשורשי היחידה מסדר n.

הגדרה. ω_n נקרא שורש יחידה פרימיטיבי.

מסקנה. מהחזקות של שורש יחידה פרימיטיבי ניתן לקבל את כל שאר שורשי היחידה.

דוגמה. עבור $n=2$ מתקיים כי $\omega_2=\cos\pi+i\sin\pi=-1$. עבור $n=4$ מתקיים $\omega_4=\cos\frac{\pi}{2}+i\sin\frac{\pi}{2}=i$ ולכן שורשי היחידה הם $\{i,-1,-i,1\}$. עבור $n=8$ מתקיים $\omega_8=\cos\frac{\pi}{4}+i\sin\frac{\pi}{4}=\frac{\sqrt{2}}{2}+i\frac{\sqrt{2}}{2}$.

עתה, ראינו כי בהנתן פולינום $P=\sum\limits_{k=0}^{n-1}a_kz^k\in V_{n-1}$ מתקיים שייצוג הערכים של P בנקודות $z_0,z_1,\ldots,z_{n-1}$ ניתן לכתיבה הבאה

$$\begin{pmatrix}P(z_0)\\ \vdots\\ P(z_{n-1})\end{pmatrix}=M\begin{pmatrix}a_0\\ \vdots\\ a_{n-1}\end{pmatrix},M=\begin{pmatrix}1 & z_0 & z_0^2 & \ldots & z_0^{n-1}\\ 1 & z_1 & z_1^2 & \ldots & z_1^{n-1}\\ \vdots & \vdots & \ddots & & \vdots\\ \vdots & \vdots & & \ddots & \vdots\\ 1 & z_{n-1} & z_{n-1}^2 & \ldots & z_{n-1}^{n-1}\end{pmatrix}$$

במקרה שלנו מתקיים $z_k=\omega_n^k$ ולכן אנו מקבלים את המטריצה

$$M=\begin{pmatrix}\omega^{0\cdot0} & \omega^{0\cdot1} & \omega^{0\cdot2} & \ldots & \omega^{0\cdot(n-1)}\\ \omega^{1\cdot0} & \omega^{1\cdot1} & \omega^{1\cdot2} & \ldots & \omega^{1\cdot(n-1)}\\ \vdots & \vdots & \ddots & & \vdots\\ \vdots & \vdots & & \ddots & \vdots\\ \omega^{(n-1)\cdot0} & \omega^{(n-1)\cdot1} & \omega^{(n-1)\cdot2} & \ldots & \omega^{(n-1)\cdot(n-1)}\end{pmatrix}$$

במילים אחרות $M_i^j=\omega^{i\cdot j}$.

הגדרה. התמרת פורייה הפוכה מסדר n DFT_n^{-1}. יהי $\begin{pmatrix}p_0\\ \vdots\\ p_{n-1}\end{pmatrix}\in\mathbb{C}^n$ ויהי $P\in V_{n-1}$ פולינום כך ש-$\begin{pmatrix}p_0\\ \vdots\\ p_{n-1}\end{pmatrix}=\begin{pmatrix}P(\omega_n^0)\\ \vdots\\ P(\omega_n^{n-1})\end{pmatrix}$ נגדיר $\begin{pmatrix}a_0\\ \vdots\\ a_{n-1}\end{pmatrix}=\mathrm{DFT}_n^{-1}\begin{pmatrix}p_0\\ \vdots\\ p_{n-1}\end{pmatrix}$. באופן מפורש, נגדיר

$$\mathrm{DFT}_n^{-1}=M^{-1}\begin{pmatrix}p_0\\ \vdots\\ p_{n-1}\end{pmatrix}=M^{-1}\begin{pmatrix}P(\omega_n^0)\\ \vdots\\ P(\omega_n^{n-1})\end{pmatrix}=\begin{pmatrix}a_0\\ \vdots\\ a_{n-1}\end{pmatrix}$$

הערה. היא מוגדרת היטב ש-DFT_n היא איזומורפיזם ובפרט הפיכה.

26.2 משפט ה-FFT

למה. $\left[M^{-1}\right]_k^m=\frac{1}{n}\left(\omega_n^{-k\cdot m}\right)$.

הערה. התמרת פורייה בדידה מעבירה מייצוג המקדמים של פולינום לייצוג הערכים שלו בשורשי היחידה מסדר n.

משפט. (משפט ה-FFT) יהי n חזקה של 2 אזי:

(i): לכל $\begin{pmatrix}a_0\\ \vdots\\ a_{n-1}\end{pmatrix}\in\mathbb{C}^n$ ניתן לחשב את $\mathrm{DFT}_n\begin{pmatrix}a_0\\ \vdots\\ a_{n-1}\end{pmatrix}$ בזמן $\mathcal{O}(n\log n)$

(ii): לכל $\begin{pmatrix}a_0\\ \vdots\\ a_{n-1}\end{pmatrix}\in\mathbb{C}^n$ ניתן לחשב את $\mathrm{DFT}_n^{-1}\begin{pmatrix}a_0\\ \vdots\\ a_{n-1}\end{pmatrix}$ בזמן $\mathcal{O}(n\log n)$

דוגמה. עבור $n=4$ מתקיים כי שורשי היחידה הם $\{1,i,-i,-1\}$ כי $\omega_4=i$ ולכן

$$M=\left(\begin{array}{cc|cc}1&1&1&1\\1&i&-1&-i\\\hline 1&-1&1&-1\\1&-i&-1&i\end{array}\right)$$

נבחין כי ארבע המטריצות הנ"ל מתקבלות אחת מהשנייה כמעט בחינם, על ידי כפל שורה או עמודה ב--1. תכונה זו נשמרת גם עבור n כללי.

הערה. את הכיוון ההפוך אפשר להסיק בצורה דומה. נסמן $Q(z)=\sum_{k=0}^{n-1}p_kz^k$ אזי $\mathrm{DFT}_n^{-1}\begin{pmatrix}p_0\\\vdots\\p_{n-1}\end{pmatrix}=\frac{1}{n}\begin{pmatrix}Q\left(\omega_n^{-1}\right)\\\vdots\\Q\left(\omega_n^{-(n-1)}\right)\end{pmatrix}$, ואז ההוכחה תהיה מאוד דומה.

לקראת הוכחת המשפט נוכיח שתי למות. המטרה שלנו היא לזהות מבנה רקורסיבי.

למה. (1) יהי n מספר זוגי ויהי $P=\sum_{k=0}^{n-1}a_kz^k\in V_{n-1}$ נגדיר $P_0(y)=\sum_{j=0}^{\frac{n}{2}-1}a_{2j}y^j$ וגם $P_1(y)=\sum_{j=0}^{\frac{n}{2}-1}a_{2j+1}y^j$ אזי $P(z)=P_0\left(z^2\right)+z\cdot P_1\left(z^2\right)$.

דוגמה. מתקיים

$$\begin{aligned}P(z)&=-z^3+7z^2+2z-4=\left(7z^2-4\right)+\left(-z^3+2z\right)\\&=\left(7z^2-4\right)+z\cdot\left(-z^2+2\right)=P_0\left(z^2\right)+z\cdot P_1\left(z^2\right)\end{aligned}$$

כאשר $P_0(y)=7y-4, P_1(y)=-y+2$.

הוכחה: (למה (1)) מתקיים

$$\begin{aligned}P_0\left(z^2\right)+z\cdot P_1\left(z^2\right)&=\sum_{j=0}^{\frac{n}{2}-1}a_{2j}z^{2j}+z\cdot\sum_{j=0}^{\frac{n}{2}-1}a_{2j+1}z^{2j}\\&=\sum_{j=0}^{\frac{n}{2}-1}a_{2j}z^{2j}+\sum_{j=0}^{\frac{n}{2}-1}a_{2j+1}z^{2j+1}\\&=\sum_{k=0}^{n-1}a_kz^k=P(z)\end{aligned}$$

כרצוי. ■ עתה נרצה לזהות מבנה רקורסיבי בתוך z^2.

למה. (2) יהי n מספר זוגי ויהי ω_n שורש יחידה פרימיטיבי מסדר n ויהי $\omega_{\frac{n}{2}}$ מסדר $\frac{n}{2}$. אזי לכל $0\le j\le\frac{n}{2}-1$ מתקיים $\left(\omega_n^{\frac{n}{2}+j}\right)^2=\left(\omega_n^j\right)^2=\omega_{\frac{n}{2}}^j$, כלומר כשמעלים את שורשי היחידה מסדר n, מקבלים פעמיים את שורשי היחידה מסדר $\frac{n}{2}$.

דוגמה. $n=4$ ו-$\omega_n=i,\omega_{\frac{n}{2}}=-1$ אזי

$$\begin{aligned}&1,i,-1,-i\\&1^2,i^2,(-1)^2,(-i)^2\\&1,-1,1,-1\end{aligned}$$

מתקיים.

הוכחה: (למה (2)) נשים לב לכך ש-$\omega_n^2=\omega_{\frac{n}{2}}$ ולכן עבור $0\le j\le\frac{n}{2}-1$ מתקיים

$$\left(\omega_n^j\right)^2=\omega_n^{2j}=\left(\omega_n^2\right)^j=\omega_{\frac{n}{2}}^j$$
$$\left(\omega_n^{\frac{n}{2}+j}\right)^2=\omega_n^{n+2j}=\omega_n^n\cdot\omega_n^{2j}=1\cdot\left(\omega_n^2\right)^j=\omega_{\frac{n}{2}}^j$$

כרצוי. ■

הוכחה: (הוכחת משפט ה-FFT) יהי $\begin{pmatrix}a_0\\ \vdots\\ a_{n-1}\end{pmatrix}\in\mathbb{C}^n$ ויהי $P=\sum\limits_{i=0}^{n-1}a_iz^i$.

מלמה 1 מתקיים כי $P(z)=P_0\left(z^2\right)+z\cdot P_1\left(z^2\right)$ כאשר $P_0(y)=\sum\limits_{i=0}^{\frac{n}{2}-1}a_{2i}y^i, P_1(y)=\sum\limits_{i=0}^{\frac{n}{2}-1}a_{2i+1}y^i$

אזי

$$\mathrm{DFT}_n\begin{pmatrix}a_0\\ \vdots\\ a_{n-1}\end{pmatrix}=\begin{pmatrix}P\left(\omega_n^0\right)\\ \vdots\\ P\left(\omega_n^{n-1}\right)\end{pmatrix}=\left(\begin{array}{c}P_0\left(\left(\omega_n^0\right)^2\right)+\omega_n^0P_1\left(\left(\omega_n^0\right)^2\right)\\ \vdots\\ P_0\left(\left(\omega_n^{\frac{n}{2}-1}\right)^2\right)+\omega_n^{\frac{n}{2}-1}P_1\left(\left(\omega_n^{\frac{n}{2}-1}\right)^2\right)\\ \hline P_0\left(\left(\omega_n^{\frac{n}{2}}\right)^2\right)+\omega_n^{\frac{n}{2}}P_1\left(\left(\omega_n^{\frac{n}{2}}\right)^2\right)\\ \vdots\\ P_0\left(\left(\omega_n^{n-1}\right)^2\right)+\omega_n^{n-1}P_1\left(\left(\omega_n^{n-1}\right)^2\right)\end{array}\right)$$

$$=\left(\begin{array}{c}P_0\left(\left(\omega_n^0\right)^2\right)\\ \vdots\\ P_0\left(\left(\omega_n^{\frac{n}{2}-1}\right)^2\right)\\ \hline P_0\left(\left(\omega_n^{\frac{n}{2}}\right)^2\right)\\ \vdots\\ P_0\left(\left(\omega_n^{n-1}\right)^2\right)\end{array}\right)+\begin{pmatrix}\omega_n^0\\ \omega_n^1\\ \vdots\\ \vdots\\ \omega_n^{n-1}\end{pmatrix}\cdot\left(\begin{array}{c}P_1\left(\left(\omega_n^0\right)^2\right)\\ \vdots\\ P_1\left(\left(\omega_n^{\frac{n}{2}-1}\right)^2\right)\\ \hline P_1\left(\left(\omega_n^{\frac{n}{2}}\right)^2\right)\\ \vdots\\ P_1\left(\left(\omega_n^{n-1}\right)^2\right)\end{array}\right)$$

$$\overset{\text{למה 2}}{=}\left(\begin{array}{c}P_0\left(\omega_{\frac{n}{2}}^0\right)\\ \vdots\\ P_0\left(\omega_{\frac{n}{2}}^{\frac{n}{2}-1}\right)\\ \hline P_0\left(\omega_{\frac{n}{2}}^0\right)\\ \vdots\\ P_0\left(\omega_{\frac{n}{2}}^{\frac{n}{2}-1}\right)\end{array}\right)+\begin{pmatrix}\omega_n^0\\ \omega_n^1\\ \vdots\\ \vdots\\ \omega_n^{n-1}\end{pmatrix}\cdot\left(\begin{array}{c}P_1\left(\omega_{\frac{n}{2}}^0\right)\\ \vdots\\ P_1\left(\omega_{\frac{n}{2}}^{\frac{n}{2}-1}\right)\\ \hline P_1\left(\omega_{\frac{n}{2}}^0\right)\\ \vdots\\ P_1\left(\omega_{\frac{n}{2}}^{\frac{n}{2}-1}\right)\end{array}\right)$$

נשים לב לכך שווקטור המקדמים של P_0 הוא בדיוק $\begin{pmatrix}a_0\\ a_2\\ \vdots\\ a_{n-2}\end{pmatrix}$ ושל P_1 הוא $\begin{pmatrix}a_1\\ a_3\\ \vdots\\ a_{n-1}\end{pmatrix}$ ולכן על פי הגדרת DFT_n נקבל כי

$$\mathrm{DFT}_{\frac{n}{2}}\begin{pmatrix}a_0\\ a_2\\ \vdots\\ a_{n-2}\end{pmatrix}=\begin{pmatrix}P_0\left(\omega_{\frac{n}{2}}^0\right)\\ \vdots\\ P_0\left(\omega_{\frac{n}{2}}^{\frac{n}{2}-1}\right)\end{pmatrix}$$

ובאותו אופן

$$\mathrm{DFT}_{\frac{n}{2}}\begin{pmatrix}a_1\\ a_3\\ \vdots\\ a_{n-1}\end{pmatrix}=\begin{pmatrix}P_1\left(\omega_{\frac{n}{2}}^0\right)\\ \vdots\\ P_1\left(\omega_{\frac{n}{2}}^{\frac{n}{2}-1}\right)\end{pmatrix}$$

ולכן, הוכחנו את הזהות הבאה:

$$\mathrm{DFT}_n\begin{pmatrix}a_0\\ \vdots\\ a_{n-1}\end{pmatrix}=\left(\frac{\mathrm{DFT}_{\frac{n}{2}}\begin{pmatrix}a_0\\a_2\\ \vdots\\ a_{n-2}\end{pmatrix}}{\mathrm{DFT}_{\frac{n}{2}}\begin{pmatrix}a_0\\a_2\\ \vdots\\ a_{n-2}\end{pmatrix}}\right)+\begin{pmatrix}\omega_n^0\\ \omega_n^1\\ \vdots\\ \vdots\\ \omega_n^{n-1}\end{pmatrix}\cdot\left(\frac{\mathrm{DFT}_{\frac{n}{2}}\begin{pmatrix}a_1\\a_3\\ \vdots\\ a_{n-1}\end{pmatrix}}{\mathrm{DFT}_{\frac{n}{2}}\begin{pmatrix}a_1\\a_3\\ \vdots\\ a_{n-1}\end{pmatrix}}\right)$$

ומכאן, כדי לחשב את $\mathrm{DFT}_n\begin{pmatrix}a_0\\ \vdots\\ a_{n-1}\end{pmatrix}$ נחשב באופן רקורסיבי את $\mathrm{DFT}_{\frac{n}{2}}\begin{pmatrix}a_1\\a_3\\ \vdots\\ a_{n-1}\end{pmatrix}, \mathrm{DFT}_{\frac{n}{2}}\begin{pmatrix}a_0\\a_2\\ \vdots\\ a_{n-2}\end{pmatrix}$ ונשתמש בזהות שהוכחנו.

נסמן ב-$T(n)$ את זמן הריצה לחישוב DFT_n, נקבל

$$T(n)\le 2T\left(\frac{n}{2}\right)+\underbrace{3n}_{\text{פעולות העתקה n פעולות כפל n פעולות חיבור n}}$$

ולכן $T(n)\le 3n\log n+\mathcal{O}(n)$ כרצוי. למה כתבנו קטן או שווה? יתכן שאפשר לעשות זאת בקצת פחות מ-$3n$. ■ דבר זה מוביל לאלגוריתם הבא לכפל פולינומים בייצוג המקדמים:

26.3 אלגוריתם מהיר לכפל פולינומים בייצוג המקדמים

קלט שני פולינומים $P,Q\in V_{n-1}$ בייצוג המקדמים

פלט הפולינום $R=PQ\in V_{2n-2}$ בייצוג המקדמים.

אלגוריתם 31 כפל פולינומים מהיר בייצוג המקדמים

---

1. אתחול: תהי m החזקה המינימלית של 2 הגדולה מ-$2n-2$ דהיינו $\min\left\{i:2n-2\le 2^i\right\}=m$. יהיו $\begin{pmatrix}b_0\\ \vdots\\ b_{n-1}\end{pmatrix},\begin{pmatrix}b_0\\ \vdots\\ b_{n-1}\end{pmatrix},\begin{pmatrix}c_0\\ \vdots\\ c_{2n-2}\end{pmatrix}$ וקטורי המקדמים של P,Q,R בהתאמה.
2. נחשב את $\begin{pmatrix}p_0\\p_1\\ \vdots\\ p_{m-1}\end{pmatrix}=\mathrm{DFT}_m\begin{pmatrix}a_0\\a_1\\ \vdots\\ a_{n-1}\\0\\ \vdots\\0\end{pmatrix}$ וגם $\begin{pmatrix}q_0\\q_1\\ \vdots\\ q_{m-1}\end{pmatrix}=\mathrm{DFT}_m\begin{pmatrix}b_0\\b_1\\ \vdots\\ b_{n-1}\\0\\ \vdots\\0\end{pmatrix}$.
3. נחשב את $\begin{pmatrix}\tilde{c}_0\\ \tilde{c_1}\\ \vdots\\ \tilde{c}_{m-1}\end{pmatrix}=\mathrm{DFT}_m^{-1}\begin{pmatrix}p_0q_0\\p_1q_1\\ \vdots\\ p_{m-1}q_{m-1}\end{pmatrix}$.
4. נחזיר את $\begin{pmatrix}\tilde{c}_0\\ \tilde{c_1}\\ \vdots\\ \tilde{c}_{m-1}\end{pmatrix}$.

טענה. (i) : האלגוריתם מחזיר את וקטור המקדמים של R.
(ii) : זמן הריצה של האלגוריתם הוא $\mathcal{O}(n\log n)$.

הוכחה: יהי ω_m שורש היחידה הפרימיטיבי מסדר m. נסמן $z_k=\omega_m^k$ לכל $0\le k\le m-1$.

(i) : מתקיים, על פי הגדרת DFT_m כי

$$\begin{pmatrix}p_0\\p_1\\ \vdots\\ p_{m-1}\end{pmatrix}=\begin{pmatrix}P(z_0)\\P(z_1)\\ \vdots\\ P(z_{m-1})\end{pmatrix},\begin{pmatrix}q_0\\q_1\\ \vdots\\ q_{m-1}\end{pmatrix}=\begin{pmatrix}Q(z_0)\\Q(z_1)\\ \vdots\\ Q(z_{m-1})\end{pmatrix}$$

ולכן

$$\begin{pmatrix} p_0q_0 \\ p_1q_1 \\ \vdots \\ p_{m-1}q_{m-1} \end{pmatrix} = \begin{pmatrix} R(z_0) \\ R(z_1) \\ \vdots \\ R(z_{m-1}) \end{pmatrix}$$

מכיוון שלפי בחירת m מתקיים $2n-2 < m$, R הוא הפולינום היחיד ממעלה $\leq m-1$ המקבל את הערכים האלה בנקודות $z_0, z_1, \ldots, z_{m-1}$ ולכן לפי הגדרת DFT_m^{-1} נקבל

$$\begin{pmatrix} \tilde{c}_0 \\ \tilde{c_1} \\ \vdots \\ \tilde{c}_{m-1} \end{pmatrix} = \mathrm{DFT}_m^{-1} \begin{pmatrix} p_0q_0 \\ p_1q_1 \\ \vdots \\ p_{m-1}q_{m-1} \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{2n-2} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

ולכן בשלב 4 של האלגוריתם נחזיר את $\begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{2n-2} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} \tilde{c}_0 \\ \tilde{c_1} \\ \vdots \\ \tilde{c}_{m-1} \end{pmatrix}$ כמתבקש.

(ii) : האלגוריתם מבצע שלוש הרצות של DFT ועוד $\mathcal{O}(m)$ פעולות, ולכן לפי משפט ה-FFT, זמן הריצה שלו הוא $\mathcal{O}(m\log m)$ אבל לפי הגדרת m $\frac{m}{2} \leq 2n-2$ ולכן $m \leq 4n-4$ ולכן $m = \mathcal{O}(n)$ ולכן זמן הריצה הוא $\mathcal{O}(n\log n)$. ■

27 תרגול 12 - פולינומים ו-DFT_n, קונבולוציה

27.1 פולינומים

הגדרה. פולינום הוא ביטוי מהצורה $P(x) = \sum\limits_{i=0}^{n-1} a_0x^i$ וניתן לייצגו לפי $\begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix}$.

טענה. בהנתן הנקודות $(x_0, y_0), \ldots, (x_{n-1}, y_{n-1})$ קיים פולינום יחיד מדרגה $\leq n-1$ כך שלכל $i \neq j, x_i \neq x_j$ כך ש-$P(x_i) = y_i$ לכל i.

הגדרה. שורש יחידה מסדר n הוא מספר מרוכב $z \in \mathbb{C}$ שמקיים $z^n = 1$. נסמן את השורש היחידה ה-i ב-ω_n^i.

טענה. $\omega_n = e^{\frac{2\pi i}{n}}$ הוא שורש יחידה וכל החזקות שלו $0, \ldots, n-1$ הן כל שאר שורשי היחידה.

הגדרה. פונקציית ה-DFT_n מוגדרת על ידי $\begin{pmatrix} P(\omega_n^{n-1}) \\ \vdots \\ P(\omega_n^{n-1}) \end{pmatrix} = \mathrm{DFT}_n \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix}$ כאשר $P = \sum\limits_{i=0}^{n-1} a_ix^i$.

משפט. קיים אלגוריתם שנסמנו FFT שמחשב את ה-DFT_n ב-$\Theta(n\log n)$. באופן דומה קיים FFT^{-1} מחשב את DFT_n^{-1} ב-$\Theta(n\log n)$.

27.2 כפל פולינומים

בעיה. יהיו שני פולינומים $p(x) = \sum\limits_{i=0}^{n-1} a_ix^i, q(x) = \sum\limits_{i=0}^{n-1} b_ix^i$. מהו $pq(x)$?

אנו יודעים כי $c_i = \sum_{i+j=n} a_i b_j$ שכן

$$\begin{aligned} p(x)\,q(x) &= \left(\sum_{i=0}^{n-1} a_i x^i\right)\left(\sum_{i=0}^{n-1} b_i x^i\right) = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} a_i x^i b_j x^j \\ &= \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} a_i b_j x^{i+j} \\ &= \sum_{i=0}^{2n-2}\left(\sum_{j=0}^{i} a_{i-j} b_j\right) x^i \end{aligned}$$

שזה לוקח $\Theta\left(n^2\right)$. האם אפשר לעשות זאת מהר יותר?

27.3 אלגוריתם מהיר לכפל פולינומים ב-$\Theta(n\log n)$

קלט $[p] = \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1}\end{pmatrix}, [q] = \begin{pmatrix} b_0 \\ \vdots \\ b_{n-1}\end{pmatrix}$.

אלגוריתם 32 אלגוריתם מהיר לכפל פולינומים

1. נחשב $\mathrm{DFT}_n\begin{pmatrix} b_0 \\ \vdots \\ b_{n-1}\end{pmatrix} = \begin{pmatrix} q(\omega_n^0) \\ \vdots \\ q(\omega_n^{n-1})\end{pmatrix}, \mathrm{DFT}_n\begin{pmatrix} a_0 \\ \vdots \\ a_{n-1}\end{pmatrix} = \begin{pmatrix} p(\omega_n^0) \\ \vdots \\ p(\omega_n^{n-1})\end{pmatrix}$.
2. נחשב $PQ = \begin{pmatrix} p(\omega_n^0)\cdot q(\omega_n^0) \\ \vdots \\ p(\omega_n^{n-1})\cdot q(\omega_n^{n-1})\end{pmatrix}$ ונחזיר $\mathrm{DFT}_n^{-1}(PQ)$.

אבל האלגוריתם הנ"ל לא טוב, שכן כדי לקבוע פולינום ממעלה $\deg pq = 2n-2$ צריך $2n-2$. על כן, נצטרך שורשי יחידה מסדר $2n-2$ לכל אחד מהפולינומים, ואז התוצאה PQ תקבע ביחידות את הפולינום.

נתקן אם כך, ונקבל את האלגוריתם הבא:

אלגוריתם 33 אלגוריתם מהיר לכפל פולינומים

1. נגדיר m להיות החזקה הקטנה ביותר של 2 שגודלה לפחות $2n-1$ ונרפד את $\begin{pmatrix} b_0 \\ \vdots \\ b_{n-1}\end{pmatrix}, \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1}\end{pmatrix}$ באפסים עד שיהיו באורך m.
2. נחשב $\mathrm{DFT}_m\begin{pmatrix} b_0 \\ \vdots \\ b_m\end{pmatrix} = \begin{pmatrix} q(\omega_m^0) \\ \vdots \\ q(\omega_m^{m-1})\end{pmatrix}, \mathrm{DFT}_m\begin{pmatrix} a_0 \\ \vdots \\ a_m\end{pmatrix} = \begin{pmatrix} p(\omega_m^0) \\ \vdots \\ p(\omega_m^{m-1})\end{pmatrix}$.
3. נחשב $PQ = \begin{pmatrix} p(\omega_m^0)\cdot q(\omega_m^0) \\ \vdots \\ p(\omega_m^{m-1})\cdot q(\omega_m^{m-1})\end{pmatrix}$ ונחזיר $\mathrm{DFT}_m^{-1}(PQ)$.
4. זמן ריצה - $\Theta(n\log n)$.

27.4 קונבולציה

הגדרה. בהנתן שני וקטורים $a = \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix}, b = \begin{pmatrix} b_0 \\ \vdots \\ b_{m-1} \end{pmatrix}$ הקונבולוציה $(a*b) = (c_0, c_1, \ldots, c_{n+m-2})$ כאשר

$$(a*b)_i = \sum_{j=0}^{i} a_j b_{i-j}$$

תכונות

1. קומוטטיביות: $a*b = b*a$.
2. אסוציאטיביות: $(a*b)*c = a*(b*c)$.

מסקנה. אלה בדיוק מקדמי ווקטור המכפלה של פולינוס המכפלה המתקבל מכפל הפולינומים הנקבעים על ידי a, b. לכן ניתן לחשב אותה ב-$\Theta(n \log n)$.

דוגמה. נגדיר $b = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, a = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 2 \end{pmatrix}$ נחשב את $c = a*b$ אזי

$$\begin{aligned} c_0 &= a_0 b_0 = 1 \\ c_1 &= a_0 b_1 + a_1 b_0 = 3 + 2 = 5 \\ c_2 &= a_0 b_2 + a_1 b_1 + a_2 b_0 = 0 + 2 \cdot 3 + 3 \cdot 1 = 9 \end{aligned}$$

האם יש דרך נוחה יותר ויזואלית? נבחין כי $\sum_{j=0}^{i} a_j b_{i-j}$ זה להציב את הווקטור a עד אינדקס i ואת הווקטור b עד האינדקס i כך שבהתחלה, אנו מציבים את a, b אחד מעל השני, הופכים את הסדר של b, וממקמים את האיבר האחרון בסדר החדש, להיות מתחת לאיבר הראשון ב-a. נסמן את האינדקסים ב-$1, \ldots, n$ אזי אנו מקבלים

$$\begin{matrix} 1 & 2 & \ldots & n \\ 1 & 2 & \ldots & n \end{matrix} \Rightarrow \begin{matrix} 1 & 2 & \ldots & n \\ n & n-1 & \ldots & 1 \end{matrix} \Rightarrow \begin{matrix} & & & 1 & 2 & \ldots & n \\ n & n-1 & \ldots & 1 & & & \end{matrix}$$

האיבר הראשון הוא כפל סקלרי של הווקטורים הנ"ל שקיבלנו כלומר $a_0 b_0$, שכן כל השאר מרופד באפסים, נסיע את b החדש איבר אחד ימינה ונקבל

$$\begin{matrix} & & & 1 & 2 & \ldots & n \\ n & n-1 & \ldots & 2 & 1 & & \end{matrix}$$

עכשיו האיבר השני הוא $a_0 b_1 + a_1 b_0$. נמשיך להסיע ובהסעה ה-j נקבל את האיבר ה-j. זו דרך ויזואלית טובה לביצוע קונבולציה. למעשה השיקוף נובע מכך שאנו מסתכלים על $-j$ וההסעה היא שאנו מסתכלים על $i-j$.

27.5 בעיית המחרוזות

קלט $P = (p_0, \ldots, p_{m-1}), S = (s_0, \ldots, s_{n-1})$ כך שלכל i מתקיים $s_i, p_i \in \{-1, 1\}$ ונניח ש-$n \le m$.

פלט D שהיא קבוצת האינדקסים k כך שבאינדקס ה-k של S מתחיל רצף של P. כלומר $D = \{k \in \{0, \ldots, n-1\} \mid \forall k \le i \le m-1 : s_i = p_i\}$.

דוגמה. $p = (1, -1, -1), q = (1, -1, -1, 1)$ במקרה זה $D = \{0\}$.

כיצד נעשה זאת? נבחין כי רצף זהה, אם"ם מכפלת כל איברים מתאימים היא 1 אם"ם הסכום של המכפלות הוא אורך הרצף, על כן נוכל לחשב את כל המכפלות הנ"ל, ואם המקדם שנקבל הוא שווה לאורך הרצף, נדע שיש התאמה. כיצד נחשב זאת? נרצה להשתמש ב-DFT_n אבל הבעיה היא שהוא מחשב מכפלות הפוך כלומר $\sum\limits_{i=0}^{k} a_i b_{k-i}$. כדי לטפל בזה, נהפוך את הסדר של אחת המחרוזות.

דוגמה. $s = (1,-1,-1,1)\,, p^R = (-1,-1,1)$ כאשר p^R הוא היפוך סדר האיברים של p אזי

$$c_0 = -1$$
$$c_1 = 0$$
$$c_2 = 3$$
$$c_3 = -1$$

ולכן c_2 מקיים את התנאי ומכאן $0,1,2$ הופעה של p ב-s. נבחין כי המקדמים הרלוונטים הם המקדמים שכוללים בתוכם את כל p.

אלגוריתם 34 אלגוריתם למציאת מופעים של מחרוזת במחרוזת אחרת

1. קלט: p, s. נגדיר p^R כך שלכל $0 \le i \le m-1$ מתקיים $p_i^R = p_{m-i-1}$.
2. נחשב את $c = \left(p^R * s\right)$.
3. נחזיר את $\left\{\underbrace{\underbrace{k}_{\text{אינדקס הסיום}} - (m-1)}_{\text{אינדקס ההתחלה}} \mid c_k = m\right\} \overset{\substack{k = l + (m-1) \\ l = k - (m-1)}}{=} \left\{l \mid c_{l+(m-1)} = m\right\} = D$.

טענה. $c_i = m$ אם"ם קיים מופע רציף של p ב-s אשר מתחיל ב-$i-(m-1)$.

הוכחה: נבחין כי

$$c_i = \sum_{j=0}^{i} p_j^R s_{i-j} = \sum_{j=0}^{i} p_{m-1-j} s_{i-j} \overset{\substack{l = m-1-j \\ j = m-1-l}}{=} \sum_{l=m-1}^{m-1-i} p_l s_{i-(m-1-l)}$$
$$= \sum_{l=m-1-i}^{m-1} p_l s_{l+(i-(m-1))}$$

אם $i > m-1$ נקבל 0 בכל l שלילי ואם $i < m-1$ אף פעם לא נקבל סכום $\le m$. לכן מספיק להסתכל על $l \ge 0$ כלומר $i \le m-1$.
מכאן $c_i = m \iff$ סכמנו בדיוק m אחדות בסכום הנ"ל $\iff$ לכל $0 \le l \le m-1$ מתקיים $p_l = s_{l+(i-(m-1))} \iff$ מתחיל רצף של p באינדקס $i-(m-1)$ ב-s. ■

חלק VII

קריפטוגרפיה ואלגוריתמים על מספרים

28 מבוא

נציג את שיטת ההצפנה של RSA וההיבטים המתמטיים והאלגוריתמים הקשורים אליה.

מהי הצפנה? נניח כי $A = \text{Alice}$ רוצה להעביר הודעה m ל-$B = \text{Bob}$. אליס רוצה להצפין את ההודעה m ולקבל הודעה m' שתוכל להעביר לבוב שיפענח אותה. הבעיה היא שבאמצע עומד Trudy The Intruder שרוצה לדעת מה אליס שלחה לבוב, ועוד הרבה רכיבים שמשבשים את ההודעה בדרך שנסמנם יחד עם Trudy ב-E.

אנו מקבלים את התרשים הבא:

$$A \to \boxed{m} \to \boxed{\text{Encryption}} \to \boxed{m'} \begin{bmatrix} E \\ \to \\ \end{bmatrix} \to \boxed{m'} \to \boxed{\text{Decryption}} \to \boxed{m} \to B$$

ראשית, נבהיר כי אנו לא נתעסק בשאר הרכיבים מלבד Trudy. נוסף על כך, אנו מציגים כאן מצב מעט מוזר - אנחנו שולחים את ההודעה המוצפנת ככה שכל העולם יכול לנסות לפענח אותה, נשמע מפוקפק. מיד נראה שלא.

נותר לקבוע כיצד להחליט מהי ההצפנה.

הגדרה. הצפנה קלאסית היא הצפנה בה ההצפנה והפענוח מתבצעים באמצעות מפתחות הצפנה ופענוח סודיים המגדירים פונקציות הצפנה ופענוח $f_E, f_D : M \to M'$ כך שלכל $m \in M$ מתקיים $m = f_D(f_E(m))$. M היא קבוצת כל ההודעות ו-M' היא קבוצת כל ההודעות ההצפנות.

כאן המפתחות סודיים וזה נשמע הגיוני, אך מסתבר שאפשר להסתדר גם ללא סודיות זו.

ב-1977 הציעו $\text{Diffie} - \text{Hellman}$ את רעיון ההצפנה הפומבית.

הגדרה. הצפנה פומבית היא הצפנה בה לכל משתמש יש מפתח הצפנה פומבי ומפתח פענוח פרטי המגדירים פונקציית הצפנה פומבית הידועה לכולם $f_E : M \to M$ כלומר $M = M'$ ופונקציית פענוח פרטי $f_D : M \to M$ כך ש-$f_E(f_D(m)) = f_D(f_E(m)) = m$.

החידוש הוא שמפתח ההצפנה זמין לכולם ואילו מפתח הפענוח הוא הפרמטר הסודי. מההגדרה, נובע שכדי לפרוץ את שיטת ההצפנה, צריך להיות מסוגלים למצוא את המקור של $f_E(m)$, קרי, את m, או אם אפשר להפוך את f_E ולקבל את f_D.

לכן, נרצה להבטיח שפונקציית הפענוח f_E קשה מאוד להפיכה ולמציאת המקור שלה. לפונקציית כאלה אנו קוראים One Way Functions - פונקציות שקלות לחישוב כיוון אחד אבל מאוד קשה להפוך אותן.

למעשה, אנחנו לא מסוגלים להוכיח שפונקציה היא אכן כזו, אלא רק לשער זאת. בעצם, מה שנותר לנו לעשות הוא למצוא פונקצית f_E, f_D שיספקו הצפנה מספיק טובה ומספיק נוחה לחישוב.

ב-1978 הוצעו שיטת הצפנה פומביות על ידי $\text{RSA}, \text{Rabin}, \text{D} - \text{H}$.

בכל השיטות $M = \{0, 1, \ldots, N-1\}$ כאשר N גדול מאוד. השיטה שנציג היא RSA והיא השיטה שתפסה לאורך השנים.

דוגמה. בהנתן $M = \{0, 1, \ldots, N-1\}$ נוכל להציע את הפונקציות הבאות

$$x \to x + 1$$

$$p \in \mathbb{P}\ x \to p \cdot x$$

האם הן מספיק טובות? ככל הנראה, ממש לא. שכן ההפוכה של הפונקציה הראשונה היא $y \to y-1$ ושל השנייה היא פשוט $y \to p^{-1}y$ כאשר p^{-1} הוא ההפכי של p מודולו N, שזאת, בקרוב נלמד לחשב באופן יעיל. נוכל להציע את השיטה הבאה,

$$x \to x^2 \mod N$$

כדי להפוך אותה, נצטרך למצוא שורש ריבועי מודולו N, שזו משימה לא מאוד קלה. למשל, לפני שנתחיל לנסות למצוא שורש, מי אמר שקיים איבר כזה? אנו מניחים ששיטת ההצפנה ידועה, ולכן ניתן להניח כי אנו מסתכלים על $x^2 \mod N$, נוכל לסמן ב-$k = x^2$, נרצה לפתור את המשוואה $y^2 \equiv k \mod n$, משוואה זו ניתן לפתור על ידי שימוש בשורש פרימיטיבי a מודולו N ואז נצטרך לפתור את המשוואה

$$ind_a(2)\, x \equiv ind_a(k) \mod \varphi(n)$$

הפלא ופלא, חישוב אינדקס של מספר זו פעולה יקרה מאוד וניתנת לביצוע בזמן ממוצע של $\Theta\left(\sqrt{N}\right)$, לכן אם N גדול מדי, זה לא פרקטי.

מי מכם שלא מכיר את המושגים שהוצגו בחלק זה של הדוגמא (שורש פרימיטיבי, φ, ind_a), זה בסדר, זזו רק העשרה.

29 שיטת ההצפנה הפומבית של RSA

הפרמטרים יהיו $N \sim 2^{1000}$.

להל"ן האלגוריתם.

אלגוריתם 35 שיטת ההצפנה הפומבית של RSA

1. נגריל שני מספרים ראשוניים גדולים $p, q \sim 2^k$ נגדיר $N = p \cdot q$ ונגדיר $M = \{0, \ldots, N-1\}$ להיות מרחב ההודעות/מרחב ההצפנות. כאשר $k \sim 500$ כלומר אלה מספרים המיוצגים בערך על ידי 500 ביטים.
2. נמצא e מספר זר ל-$\varphi(pq) = (p-1)(q-1)$ כאשר $e \sim \text{poly}(N)$
3. נמצא $d \equiv e^{-1} \pmod{(p-1)(q-1)}$ כלומר $0 \le d < (p-1)(q-1)$ כך ש-$ed \equiv 1 \pmod{(p-1)(q-1)}$.
4. נפרסם את (N, e) כמפתח הצפנה פומבי, ונשמור את d כמפתח פענוח פרטי.
5. פונקציית ההצפנה: $f_E : M \to M$ המוגדרת על ידי $f_E(x) = x^e \mod N$.
6. פונקציית הפענוח: $f_D : M \to M$ המוגדרת על ידי $f_D(y) = y^d \mod N$.

נבחין כי על מנת לפרוץ את השיטה מספיק לדעת את p, q, או במילים אחרות, לפרק את N לגורמים ראשוניים. זו משימה קשה. למעשה, אנחנו לא מסוגלים להוכיח ששבירת RSA שקולה לפירוק של מספר לגורמים ראשוניים. למרות שזו משימה קשה, באמצעות מחשבים קוואנטיים ניתן לשבור איתה, בפרט, באמצעות האלגוריתם של "Shor", שמסוגל לחשב כפל של ווקטור במטריצה אורתוגונלית, שהיא למעשה מטריצת ה-FFT בזמן $\log^2 n$, שזה מאוד מהיר.

הערה. אם נרצה לתקשר אחד עם השני - כל אחד מאיתנו צריך להריץ את האלגוריתם, שכן הרצה יחידה מאפשרת שליחת הודעה אלינו ולא ממנו הלאה.

29.1 שימושים של שיטת הצפנה פומבית

1. "ספר טלפונים" של מפתחות הצפנה פומביים. נשמור מפתח פרטי לכל ערוץ קשר ונפרסם את המפתח הפומבי, ככה נוכל לתקשר עם כולם בצורה יעילה.

2. "חתימה דיגיטלית". אם A רוצה לחתום על הודעה m, היא מפרסמת את הזוג $(m, f_D(m))$ כך שכל אחד יכול לוודא כי $f_E(f_D(m)) = m$. חתימה זו היא הבסיס לתורת המטבעות הקריפטוגרפיים.

שאלות מתמטיות

1. למה קיימים p, q כאלה?
2. למה קיים e כזה?
3. למה קיים d כזה?
4. למה לכל $x \in M$ מתקיים $f_D(f_E(x)) = x = f_E(f_D(x))$?

שאלות אלגוריתמיות

1. איך ניתן למצוא p, q באופן יעיל? (פולינומי ב-k) איך נוודא שהמספרים שמצאנו ראשוניים (אלגוריתם מילר-רבין)
2. אין למצוא e באופן יעיל?
3. איך למצוא d באופן יעיל?
4. האם ניתן לחשב את f_E, f_D באופן יעיל?

החלק המתמטי מבוסס על תורת המספרים האלמנטרית ברובה והרבה על אריתמטיקה מודולרית.

30 תורת המספרים האלמנטרית

30.1 אריתמטיקה מודולרית

טענה. בהנתן שני מספרים טבעיים $a, b \neq 0$ ניתן לחלק את a ב-b עם שארית, כלומר לרשום $a = qb + r$ עבור $a \leq r < b$. הדרך לרשום היא יחידה, q נקראת המנה ו-r נקראת השארית.

הגדרה. בהנתן מספר $b \geq 2$ נגדיר פונקציה $\bmod b : \mathbb{N} \to \{0, \ldots, b-1\}$ באופן הבא. עבור $a \in \mathbb{N}$ נחלק את a ב-b עם שארית כלומר $a = qb + r$ ונגדיר $a = r$ $(\bmod b)$.

הערה. מסיבות היסטוריות נרשם זאת $r = (\bmod b)\, a$.

אלגוריתמית בהנתן a ו-b ניתן לחשב את $a \pmod b$ באופן יעיל - בזמן פולינומי באורך הייצוגים.

צורת רישום נרשום $a \equiv c \mod b$ כדי להגיד ש-$a \pmod b = c \pmod b$.

עובדה $a \equiv c \mod b$ אם"ם $a - c$ מתחלק ב-b.

תכונות

לכל $a, b, c \neq 0$

1. $(a+c) \pmod b = a \pmod b + c \pmod b$.
2. $ac \pmod b = (a \pmod b \cdot c \pmod b) \pmod b$.

דוגמה. נחשב $(2022)^{22} \pmod{11}$. נבחין כי $2022 = 2000 = 20 \cdot 100 = 9 \pmod{11}$ ולכן

$$(2022)^{22} \equiv 9^{22} = (81)^{11} \equiv 4 \cdot 4^{10} \equiv 4 \cdot (16)^5 \equiv 4 \cdot 5^5$$

הגדרות

הגדרה. <u>המחלק המשותף המקסימלי</u> של a,b הוא המספר הגדול ביותר המחלק גם את a וגם את b. נסמן אותו ב-$\gcd(a,b)$.

דוגמה. $\gcd(40,64)=8$.

עובדה ניתן לחשב את $\gcd(a,b)$ באופן יעיל, כלומר בזמן פולינומי באורך הייצוגים של a,b.

הגדרה. p הוא מספר ראשוני אם p מחלק רק בעצמו וב-1.

טענה. (פירוק לגורמים ראשוניים) כל מספר טבעי a ניתן להצגה יחידה כמכפלת חזקות של מספרים ראשוניים שונים $a=p_1^{k_1}\cdot\ldots\cdot p_n^{k_n}$ והפירוק יחיד עד כדי סדר, כאשר $p_1,\ldots,p_n\in\mathbb{N}$ ראשוניים ו-$k_1,\ldots,k_n\in\mathbb{N}$.

טענה. a,b הינם זרים אם"ם אין מספר ראשוני המשותף בפירוק של שניהם לגורמים ראשוניים.

דוגמה. $40=2^3\cdot5, 21=3\cdot7$ ו-$\gcd(40,21)=1$ ולכן הם זרים.

משפט. (צ'בישב, התפלגות הראשוניים) עבור מספר $a\in\mathbb{N}$ נגדיר $\pi(a)=|\{2\le p\le a \mid p \text{ ראשוני}\}|$ אזי $\pi(a)\sim\frac{a}{\ln a}$, כלומר $\lim_{n\to\infty}\frac{\pi(a)}{\frac{a}{\ln a}}=1$.

שאלה כיצד נבצע את השלב הראשון באלגוריתם של RSA?

פתרון. על סמך המשפט, נוכל לבצע את השלב הראשון באלגוריתם RSA באופן הבא. נגריל מספר בן k ביטים (על ידי הגרלת k ביטים בנפרד באופן ב"ת) ונבדוק באמצעות משפט מילר-רבין האם המספר שהגרלנו ראשוני. לפי המשפט, הסיכוי להגריל ראשוני הוא $\sim\frac{1}{\ln 2^k}=\frac{1}{k\ln 2}$ ולכן לאחר מספר קבוע של ניסיונות נמצא ראשוני בסיכוי גדול. מכיוון ש-k הוא 500, בהסתברות מאוד גבוהה נקבל ש-10 הביטים המשמעותיים ביותר הם אפסים ולכן הוא יהיה לפחות מסדר גודל של 2^{490}, בהסתברות גבוהה.

נוכל לבצע גם את השלב ה-2 באלגוריתם RSA באופן הבא. נגריל e ונבדוק האם הוא ראשוני. אחרי $O(k)$ נמצא ראשוני בסיכוי גבוה. מדוע הוא יהיה זר ל-$(p-1)(q-1)$? בסיכוי גבוה הוא לא יחלק את $(p-1)(q-1)$ ולכן יהיה זר לו. נפרק את המספרים הנ"ל לגורמים ראשוניים, אזי יש לו לכל היותר $\log((p-1)(q-1))$ גורמים ראשוניים שזה לכל היותר 1000. לכן בסבירות גבוהה מאוד, נקבל ראשוני שלא חלק מהפירוק שלהם.

נרשום את השלבים שנותרו לנו לבנייה מלאה של האלגוריתם:

1. ראינו כי את המספרים הראשוניים p,q בשלב 1 של RSA ניתן למצוא באופן יעיל בהסתמך על משפט התפלגות הראשוניים $\pi(a)\sim\frac{a}{\ln a}$, ולכן אנו זקוקים לאלגוריתם יעיל (פולינומי באורך הייצוג של המספר), הבודק האם מספר נתון הוא ראשוני.
2. גם שלב זה מתבסס על משפט צפיפות הראשוניים (ראינו).
3. ראינו בתרגול כי ניתן לבצע אותו באמצעות אלגוריתם אוקלידס המורחב.

נמשיך לפתח כלים מתמטיים שיעזרו לנו לפתח אלגוריתם לבדיקת ראשוניות של מספר וגם להוכיח כי פונקציות ההצפנה והפענוח הופכיות אחת לשנייה.

למה. (1), ללא הוכחה.

(i): a מתחלק ב-b אס"ם כל גורם ראשוני שמופיע בפירוק של b לגורמים ראשוניים גם בפירוק של a ולפחות באותה החזקה שבה הוא הופיע בפירוק של b.

(ii): אם a,c זרים ל-b אזי $a\cdot c$ וגם $a\cdot c \pmod b$ זרים ל-b

(iii): אם a זר ל-b אבל $b\mid a\cdot c$ אזי $b\mid c$.

דוגמה. $a=72, b=18$ אזי $b\mid a$ וגם $a=2^3\cdot3^2, b=3^2\cdot2$.

מסקנה. יהי a זר ל-b ויהיו $1\le c,d\le b-1$ כך שמתקיים $a\cdot c\equiv a\cdot d \mod b$ אזי $c=d$.

הוכחה: מכיוון ש-$a \cdot c \equiv a \cdot d \mod b$ מתקיים $ac - ad = a(c-d)$ מתחלק ב-b. על פי ההנחה, a זר ל-b ולכן לפי חלק (iii) של למה (1), מתקיים ש-$c-d$ מתחלק ב-b, על פי ההנחה, $1 \le c, d \le b-1$ ולכן $|c-d| < b$ ולכן מהיות $b \mid c-d$ אנו מקבלים כי $c - d = 0$ ולכן $c = d$. ■

30.2 מבנים אלגבריים ומשפט אוילר

הגדרה. יהי n מספר טבעי. נגדיר $\mathbb{Z}_n = \{0, 1, \dots, n-1\}$. נגדיר $\mathbb{Z}_n^* = \{0 < a < n \mid \gcd(a, n) = 1\}$. נגדיר $\varphi(n) = |\mathbb{Z}_n^*|$ הנקראת <u>פונקציית אוילר (Euler's Totient Function)</u>.

דוגמה. $n = 6$ אזי $\mathbb{Z}_6^* = \{1, 5\}$ ולכן $\varphi(6) = 2$. עבור $n = 12$ מתקבל $\mathbb{Z}_{12}^* = \{1, 5, 7, 11\}$ ולכן $\varphi(12) = 4$.

למה. (2)

(i) : יהי p מספר ראשוני אזי $\varphi(p) = p - 1$.

(ii) : יהיו p, q שני מספרים ראשוניים שונים אזי $\varphi(p \cdot q) = \varphi(p) \cdot \varphi(q) = (p-1)(q-1)$. בפרט $\varphi(a \cdot b) = \varphi(a) \cdot \varphi(b)$ לכל $\gcd(a, b) = 1$.

הוכחה: (i) : יהי p ראשוני אזי $\mathbb{Z}_p^* = \{1, \dots, p-1\}$ ולכן $\varphi(p) = |\mathbb{Z}_p^*| = p - 1$.

(ii) : נסמן $B = \{0 < b < pq \mid \gcd(b, pq) > 1\}$ אזי $B \dot\cup \mathbb{Z}_p^* = \{1, 2, \dots, pq-1\}$ ולכן

$$\varphi(pq) = |\mathbb{Z}_{pq}^*| = pq - 1 - |B|$$

B מכילה את כל המספרים הקטנים מ-pq והגדולים מאפס שמתחלקים או ב-p או ב-q. לכן

$$B = \{p, 2p, \dots, (q-1) \cdot p\} \dot\cup \{q, 2q, \dots, (p-1) q\}$$

הקבוצות הנ"ל זרות, שכן אחרת היינו מקבלים איבר מהצורה $kp = lq$ אך מכיוון ש-$\gcd(p, q) = 1$ נקבל כי $p \mid l$ ולכן $pq < lq$ ולכן pq בקבוצה. סתירה. מכאן

$$|B| = (q-1) + (p-1)$$

ולכן

$$\varphi(pq) = pq - 1 - |B| = pq - p - q + 1 = (p-1)(q-1)$$

כרצוי. ■

הגדרה. יהי $n \in \mathbb{N}$ ויהי a זר ל-n. נגדיר פונקציה f_a על מספרים טבעיים, הפועלת באופן הבא. עבור $b \in \mathbb{N}$ נגדיר $f_a(b) = a \cdot b \mod n$.

מסקנה. נשים לב כי אם $b \in \mathbb{Z}_n^*$ אזי $f_a(b) \in \mathbb{Z}_n^*$. זה נכון משתי סיבות. קודם כל, עבור $0 \le c \le n-1$ וגם מכיוון ש-a זר ל-n על פי ההנחה ו-$b \in \mathbb{Z}_n^*$ אזי $a \cdot b \in \mathbb{Z}_n^*$ בהסתמך על חלק 2 של למה (1). מכאן נוכל להביט על הצמצום שלה $f_a : \mathbb{Z}_n^* \to \mathbb{Z}_n^*$.

דוגמה. עבור $n = 12$ מתקיים $\mathbb{Z}_{12}^* = \{1, 5, 7, 11\}$ ו-$a = 5$ אזי

$$f_a(1) = 5 \cdot 1 = 5$$
$$f_a(5) = 5 \cdot 5 = 1$$
$$f_a(7) = 5 \cdot 7 = 11$$
$$f_a(11) = 5 \cdot 11 = 7$$

זה נראה כאילו היא חח"ע ועל.

למה. (מרכזית) יהי n מספר טבעי ויהי a זר ל-n אזי $f_a : \mathbb{Z}_n^* \to \mathbb{Z}_n^*$ הינה חח"ע ועל.

הוכחה: מכך ש- $f_a : \mathbb{Z}_n^* \to \mathbb{Z}_n^*$ מספיק להוכיח ש-f_a חד חד ערכית. יהיו $b, c \in \mathbb{Z}_n^*$ כך ש-$f_a(b) = f_a(c)$. נוכיח כי $b = c$. על פי הגדרת f_a מתקיים כי $a \cdot c \equiv a \cdot b \mod n$ אך מהיות $\gcd(a,n) = 1$ לפי המסקנה מלמה 1 מתקיים כי $b = c$. ■

מסקנה. (מהלמה המרכזית) יהי n מספר טבעי ויהי $e \in \mathbb{Z}_n^*$ אזי קיים $d \in \mathbb{Z}_n^*$ כך ש-$de = ed = 1 \mod n$.

הוכחה: נתבונן בפונקציה $f_e : \mathbb{Z}_n^* \to \mathbb{Z}_n^*$ המוגדרת על ידי $f_e(x) = ex (\mod n)$. לפי הלמה המרכזית, f_e הינה על $\mathbb{Z}_n^*$, מכיוון ש-$1 \in \mathbb{Z}_n^*$ קיים d כך ש-$f_e(d) = 1$ שזה בדיוק אומר ש-$ed = 1 (\mod n)$. ■

משפט. (אוילר) יהי n מספר טבעי ויהי a זר ל-n אזי $a^{\varphi(n)} \equiv 1 \mod n$.

דוגמה. $n = 12, a = 5$ אזי $\varphi(n) = 4$, נחשב

$$5^4 = 25^2 \equiv 1^2 = 1 \mod 12$$

הוכחה: (משפט אוילר) נגדיר $f_a : \mathbb{Z}_n^* \to \mathbb{Z}_n^*$ על ידי $f_a(x) = ax (\mod n)$ לפי הלמה המרכזית f_a היא חח"ע ועל. לכן

$$\prod_{x \in \mathbb{Z}_n^*} x = \prod_{x \in \mathbb{Z}_n^*} f_a(x) = \prod_{x \in \mathbb{Z}_n^*} (ax) (\mod n)$$

לכן

$$\left(\prod_{x \in \mathbb{Z}_n^*} x\right) (\mod n) = \left(\prod_{x \in \mathbb{Z}_n^*} (ax) (\mod n)\right) (\mod n)$$

לפי תכונות פונקציית המודולו מתקיים

$$\begin{aligned}\left(\prod_{x \in \mathbb{Z}_n^*} (ax) (\mod n)\right) (\mod n) &= \left(\prod_{x \in \mathbb{Z}_n^*} ax\right) (\mod n) \\ &= \left(a^{\varphi(n)} \cdot \prod_{x \in \mathbb{Z}_n^*} x\right) (\mod n) \\ &= \left(\left(a^{\varphi(n)} (\mod n)\right) \cdot \left(\prod_{x \in \mathbb{Z}_n^*} x\right) (\mod n)\right) (\mod n)\end{aligned}$$

נסמן $A = a^{\varphi(n)} (\mod n)$ נשים לב כי $0 \le A \le n-1$ ו-A זר ל-n לפי למה 1 כלומר $A \in \mathbb{Z}_n^*$. נסמן גם $B = \left(\prod_{x \in \mathbb{Z}_n^*} x\right) (\mod n)$, באופן דומה, $B \in \mathbb{Z}_n^*$. מהפיתוחים הקודמים:

$$B = A \cdot B (\mod n)$$

נרשום את השוויון שקיבלנו $B = AB (\mod n)$ כ-$1 \cdot B = A \cdot B (\mod n)$ מתקיים כי B זר ל-n כי $B \in \mathbb{Z}_n^*$ ו-$1 \le A \le n-1$ ולכן לפי המסקנה מלמה 1, מתקיים $A = 1$ כלומר $a^{\varphi(n)} \equiv 1 \mod n$. ■

מסקנה. (המשפט הקטן של פרמה) יהי p מספר ראשוני ונניח כי a לא מתחלק ב-p אזי $a^{p-1} \equiv 1 \mod p$.

הוכחה: נבחר $n=p$ אזי $\varphi(p)=p-1$ ולכן ממשפט אוילר $a^{\varphi(p)}=a^{p-1}\equiv 1 \mod p$. ■

משפט. לכל $x\in M$ מתקיים $x=f_E(f_D(x))=f_D(f_E(x))$.

הוכחה: לפי הגדרת פונקציית ההצפנה והפענוח מתקיים

$$\begin{aligned} f_D(f_E(x)) &= f_D(x^e \mod N)=(x^e \mod N)^d \mod N=(x^e)^d \mod N \\ &= x^{e\cdot d} \mod N = x^{d\cdot e} \mod N \end{aligned}$$

ובאופן דומה $f_E(f_D(x))=x^{d\cdot e} \mod n$.

לפי דרך פעולתו של האלגוריתם של RSA, מתקיים $de\equiv 1 \mod (p-1)(q-1)$ או באופן שקול, $de=a\cdot(p-1)(q-1)+1$ כאשר $a\in\mathbb{N}$.

לכן כדי להוכיח את המשפט, מספיק לוודא כי $x^{a(p-1)(q-1)+1}\equiv x \mod N$ לכל $a\in\mathbb{N}$.

נחלק למקרים:

(i) : x זר ל-N אזי ממשפט אוילר $x^{\varphi(N)}=x^{(p-1)(q-1)}\equiv 1 \mod N$ ולכן בחישוב מודולו N מתקיים

$$x^{a(p-1)(q-1)+1}=x\cdot x^{a\cdot(p-1)(q-1)}=x\cdot\left(x^{(p-1)(q-1)}\right)^a\equiv x\cdot 1^a=x \mod N$$

(ii) : x מתחלק ב-p אבל לא מתחלק ב-q. במקרה זה נשתמש במשפט הקטן של פרמה ונראה כי $x^{q-1}\equiv 1(\mod q)$. לכן בחישוב מודולו q נקבל כי

$$\begin{aligned} x^{a\cdot(p-1)(q-1)+1} &= x\cdot x^{a\cdot(p-1)(q-1)}=x\cdot\left(x^{q-1}\right)^{a(p-1)} \\ &\equiv x\cdot 1^{a(p-1)}=x \mod q \end{aligned}$$

נסמן $y=x^{a\cdot(p-1)(q-1)+1}$ הוכחנו כי $y\equiv x \mod q$. באופן שקול $q\mid y-x$. מצד שני, על פי ההנחה, x מתחלק ב-p ולכן גם y מתחלק ב-p ולכן גם $y-x$ מתחלק ב-p. לכן $y-x$ מתחלק גם ב-q וגם ב-p ולכן מתחלק ב-$N=pq$ ולכן

$$x^{a\cdot(p-1)(q-1)}=y\equiv x \mod N$$

(iii) : x מתחלק ב-q אבל לא מתחלק ב-p. ניתן לטפל באופן דומה.

(iv) : x מתחלק גם ב-p וגם ב-q. במקרה זה x מתחלק ב-$N=pq$. מכיוון ש-$x\in\{0,1,\dots,pq-1\}$ מתקיים $x=0$. במקרה הז השוויון המבוקש מתקיים באופן טריוויאלי. ■

31 האלגוריתם של מילר ורבין $(\mathrm{Miller-Rabin})$

קלט מספר טבעי n.

פלט "n ראשוני" או "n לא ראשוני".

נרצה למצוא אלגוריתם הפועל באופן יעיל, פולינומי ב-$\log n$.

כיצד נקבע ראשוניות? על פי המשפט הקטן של פרמה, אם n ראשוני אזי לכל a שלא מתחלק ב-n ובפרט לכל $1\le a\le n-1$ מתקיים $a^{n-1}\equiv 1 \mod n$.

טענה. אם $a^{n-1}\equiv 1 \mod n$ אזי בוודאות $\gcd(a,n)=1$.

הוכחה: קיים k טבעי כך ש-$a^{n-1}-1=k\cdot n$ ולכן $a^{n-1}=1+kn$ ומכאן אם $t\mid n$ אזי $t\mid kn$ ולכן $t\nmid kn+1$ ולכן $\gcd(a,n)=\gcd\left(a^{n-1},n\right)=1$. ■ על כן, הצעד הבסיס של האלגוריתם של מילר ורבין:

נבחר a מקרי בין 2 ל-$n-1$. נבדוק האם $a^{n-1}\equiv 1 \mod n$. נשים לב לכך שאם n ראשוני, השוויון הזה תמיד מתקיים לפי המשפט הקטן של פרמה. מתברר שכאשר n אינו ראשוני, אזי לפחות חצי מהמספרים $1\le a\le n-1$ מקיימים $a^{n-1}\not\equiv 1 \mod n$ ולכן הצעד הבסיסי יזהה שהוא אינו ראשוני בסיכוי לפחות $\frac{1}{2}$. כלומר אנו מקבלים שאיטרציה אחת של מילר רבין היא מהצורה:

אלגוריתם 36 איטרציה אחת של מילר-רבין

---

1. נבחר בצורה אחידה $a\in[m-1]$.
2. אם $a^{m-1}\equiv 1 \mod m$ נחזיר "Prime".
3. אחרת נחזיר "Composite".

למה. יהי $m\in\mathbb{N}$ ו-$B_m=\left\{a\in\mathbb{Z}_m^*\mid a^{m-1}\not\equiv 1 \mod m\right\}$ אזי אם $B_m\neq\emptyset$ אזי $|B_m|\ge\frac{|\mathbb{Z}_m^*|}{2}$.

הוכחה: יהי $m\in\mathbb{N}$ כך ש-$B_m\neq\emptyset$. יהי $b\in B_m$. נבחין כי מהלמה הקודמת, נובע כי $b\in\mathbb{Z}_m^*$ ולכן $B_m\subseteq\mathbb{Z}_m^*$.
נגדיר אם כך $f_b:\mathbb{Z}_m^*\to\mathbb{Z}_m^*$ על ידי

$$f_b(a)=a\cdot b(\mod m)$$

לכל $a\in\mathbb{Z}_m^*$. ראינו כי f_b היא פרמוטציה של $\mathbb{Z}_m^*$.
נראה כי לכל $a\in\mathbb{Z}_m^*\backslash B_m$ מתקיים כי $f_b(a)\in B_m$, ומכיוון ש-f_b היא העתקה חח"ע ועל, ינבע כי

$$|\mathbb{Z}_m^*|-|B_m|=|\mathbb{Z}_m^*|-|B_m\cap\mathbb{Z}_m^*|=|\mathbb{Z}_m^*\backslash B_m|\le|B_m|$$

ולכן $|B_m|\ge\frac{1}{2}|\mathbb{Z}_m^*|$.
מהגדרת B_m ו-$a\in\mathbb{Z}_m^*\backslash B_m$, מתקיים כי $a^{m-1}\equiv 1 \mod m$ ואילו מהיות $b\in B_m$ $b^{m-1}\not\equiv 1 \mod m$ ולכן

$$f_b(a)^{m-1}\equiv(a\cdot b)^{m-1}=a^{m-1}\cdot b^{m-1}\equiv b^{m-1}\not\equiv 1 \mod m$$

ולכן $f_b(a)\in B_m$. ■

טענה. יהי $m\in\mathbb{N}$ ו-$B_m=\left\{1\le a\le m-1\mid a^{m-1}\not\equiv 1 \mod m\right\}$ אזי אם $B_m\neq\emptyset$ אזי $|B_m|\ge\frac{m-1}{2}$.

הוכחה: נסמן $Y=\{1\le a\le m-1\mid\gcd(a,m)>1\}$ אזי $\mathbb{Z}_m^*[]\cup\cdot Y=\mathbb{Z}_m$ ולכן

$$B_m=(B_m\cap Y)[]\cup\cdot(B_m\cap\mathbb{Z}_m^*)$$

עתה, נבחין כי אם $a\in Y$ אזי $a^{m-1}\not\equiv 1 \mod m$ שכן אחרת מתכונות המודולו ינבע ש-$\gcd(a,m)=1$. לכן $a\in B_m$ ולכן מכאן $Y\subseteq B_m$

$$\begin{aligned}|B_m|&=|B_m\cap Y|+|B_m\cap\mathbb{Z}_m^*|\\&=|Y|+\left|\left\{a\in\mathbb{Z}_m^*\mid a^{m-1}\not\equiv 1 \mod m\right\}\right|\\&\ge|Y|+\frac{|\mathbb{Z}_m^*|}{2}=(m-1-|\mathbb{Z}_m^*|)+\frac{|\mathbb{Z}_m^*|}{2}\\&=m-1-\frac{|\mathbb{Z}_m^*|}{2}\ge m-1-\frac{m-1}{2}=\frac{m-1}{2}\end{aligned}$$

כרצוי. ■

משפט. יהי $m \in \mathbb{N}$. אם m ראשוני, אזי מילר-רבין מחזיר "ראשוני". אם m פריק כך ש-$B_m \neq \emptyset$ אזי מילר-רבין מחזיר "פריק" בהסתברות לפחות $\frac{1}{2}$.

הוכחה: אם המספר ראשוני, אזי ממשפט פרמה $a^{m-1} \equiv 1 \mod m$ לכל $a \in [m-1]$. מצד שני, אם m פריק כך ש-$B_m \neq \emptyset$ אזי מילר רבין יחזיר "פריק" אם ורק אם בחרנו $a \in [m-1]$ כך ש-$a^{m-1} \equiv 1 \mod m$.

בטענה הקודמת, ראינו כי $|B_m| \geq \frac{m-1}{2}$ ולכן לא יותר מחצי האיברים יובילו אותנו למסקנה "ראשוני", ולכן ההסתברות שנקבל "פריק" היא לפחות $\frac{1}{2}$. ■ אם נרצה הסתברות גבוהה יותר לקבלת התשובה הנכונה, נצטרך לבצע מספר איטרציות (בבחירה בלתי תלויה) של האלגוריתם ולהחזיר "פריק" אם לפחות אחת מהאיטרציות החזירה "פריק". אם המספר ראשוני, נחזיר "ראשוני" בהסתברות 1. אם נריץ k איטרציות עבור מספר פריק m עם $B_m \neq \emptyset$, אזי ההסתברות להחזרת "ראשוני" אינה גדולה מ-$\frac{1}{2^k}$.

הערה. יש מספרים אשר אינם ראשוניים, ומקיימים את התנאי לכל $1 \leq a \leq n-1$, כלומר $B_n = \emptyset$. האלגוריתם המלא שלא מילר ורבין מטפל במקרה זה עם תוספת קטנה.

הגדרה. יהי $m \in \mathbb{N}$ פריק. m נקרא <u>מספר קרמייקל (Charmichael)</u> כאשר $B_m = \emptyset$.

31.1 זמן ריצה

יהי $b = \lceil \log_2 m \rceil$ מספר הביטים של m. נבחין כי מספר הביטים של $a, m-1$ הם לא יותר מ-b. אנו כבר יודעים שנוכל לבצע העלאה בחזקה ופעולות מודולוריות בזמן פולינומיאלי ב-b, אז כל הצעדים מלבד הדגימה, נעשות בזמן פולינומיאלי.

אם נרצה לדייק, אנו מחשבים את $a^{m-1} \pmod m$ על ידי שימוש בהעלאה בריבוע בצורה רפטטיבית, וביצוע פעולת מודולו לאחר כל איטרציה, כדי לוודא שמספר הביטים לא גדל יותר מדי. נבחין כי מספר הביטים של a^2 הוא לכל היותר $2b$ ולכן בכל איטרציה אנו מבצעים פעולות על מספרים עם $O(b)$ ביטים. עבור שימוש במימוש הנאיבי של כפל ומודולו שראינו, אנו מעלים בריבוע את המספר $O(b)$ הפעמים, ומבצעים בכל פעם $O(b^2)$ פעולות, ולכן העלות הכוללת היא $O(b^3)$.

עלינו לנתח את עלותה של ההגרלה. נרצה פשוט לדגום בצורה ראנדומלית על ידי הגרלה של b ביטים בצורה בלתי תלויה, כך שידרשו $O(b)$ פעולות. יחד עם זאת, במקרה זה, יתכן כי $a \geq m$. כיצד נפתור זאת?

נבחין כי על מנת ש-$a \geq m$ צריך שהביט הגדול ביותר ב-a יהיה 1 וזה קורה בהסתברות $\frac{1}{2}$. על כן, נוכל לדגום את המספרים שוב ושוב עד שנקבל מספר קטן מ-m. בתוחלת, זה יקח 2 ניסיונות ובהסתברות מאוד גבוהה, לא נצטרך לבצע יותר ממספר קבוע של ניסיונות. נוכל גם לומר שאיטרציה "נכשלת" אם היא דגמה מספר גדול ולהתעלם מהפלט. במקרה זה, אם נריץ את האלגוריתם על מספר פריק שאינו פריק קרמייקל, בכל איטרציה יש לנו לכל היותר הסתברות $\frac{1}{2}$ להיכשל ואם זה לא קורה, אז לא יותר מהסתברות $\frac{1}{2}$ להחזרת "ראשוני". סך הכל, נקבל כי ההסתברות של אי החזרת "פריק" אינה גדולה מ-$\frac{1}{4}$. על כן, אם נריץ את האלגוריתם k פעמים ההסתברות להחזרת "ראשוני" סך הכל, לא תהיה גדולה מ-$\frac{1}{4^k}$.

31.2 אלגוריתם מילר רבין המלא

באלגוריתם הקודם, לא היינו מסוגלים לטפל במספרי קרמייקל. נציע קריטריון נוסף באמצעותו נרחיב את הדרך לבדיקת ראשוניות. ליתר דיוק, נגדיר פעולה $\text{Witness}(a,n)$ שתחזיר אמת אם"ם a הוא "עד" לכך ש-n פריק, או במילים אחרות, באמצעות עובדה זו נוכל להוכיח כי n פריק, באופן שמיד נראה.

לפני כן, נציג מושג חדש מתורת המספרים - שורש ממינוס אחת.

טענה. יהי p ראשוני, אזי הפתרונות היחידים למשוואה $x^2 \equiv 1 \mod p$ הם $-1, 1$ בלבד.

הוכחה: מההנחה $p \mid x^2 - 1 = (x+1)(x-1)$. מהיות p ראשוני מתקיים כי $p \mid x+1$, כלומר $x \equiv -1 \mod p$ או $p \mid x-1$ כלומר $p \equiv 1 \mod p$. כרצוי. ■

מסקנה. אם קיים פתרון $x \notin \{-1, 1\}$ למשוואה $x^2 \equiv 1 \mod n$ אזי n פריק.

דבר זה מוביל אותנו אל האלגוריתם הבא:

Algorithm 37 Witness (a, n)

1 : Let u be such that u is odd and $n-1 = 2^t u$ for $t \geq$
1 # assume n is odd, since otherwise it is trivially composite.
2 : define $x_0 = a^u \mod n$
3 : for $i = 0$ to t : # compute $a^{n-1} \mod n$
4 : $\quad x_i = x_{i-1}^2 \mod n$
5 : $\quad$ if $x_i == 1$ and $x_{i-1} \neq 1$ and $x_{i-1} \neq -1$: # non trivial solution
6 : $\qquad$ return True
7 : if $x_t \neq 1$: # contradicition to Fermat's little theorem
8 : $\quad$ return True
9 : return False

אנו מבצעים את הפירוק בשלב 1 כדי שנוכל להעלות בצורה ישירה בריבוע את x_0 ולהגיע בסוף ל-$a^{n-1} \mod n$, ובצורה זו להרוויח משני הקריטריונים לראשוניות שראינו. בנוסף, אם האלגוריתם מחזיר אמת משלב 6, זה מהקריטריון החדש, ואם זה משלב 8 זה מהקריטריון המקורי.

משפט. יהי n מספר אי זוגי ופריק. אזי מספר העדים לפריקותו הוא לפחות $\frac{n-1}{2}$.

הוכחה: ראשית נבחין כי כל איבר a שאינו עד מקיים כי $a \cdot a^{n-2} = a^{n-1} \equiv 1 \mod n$ ולכן $a \in \mathbb{Z}_n^*$.

נוכיח כי כל האיברים שאינם עדים מוכלים בתת קבוצה $B \subsetneqq \mathbb{Z}_n^*$ הסגורה לכפל. שכן אז ממשפט לגראנז' ינבע כי $|B| \mid |\mathbb{Z}_n^*|$ ולכן בפרט $\frac{n-1}{2} \leq \frac{|\mathbb{Z}_n^*|}{2} \leq |B|$. לכן קבוצת כל הלא עדים בגודל קטן מ-$\frac{n-1}{2}$ ולכן המשלימה שלה, קבוצת כל העדים, בגודל לפחות $\frac{n-1}{2}$.

נמצא את B.

ראשית, נניח כי קיים $x \in \mathbb{Z}_n^*$ כך ש-$x^{n-1} \not\equiv 1 \mod n$, כלומר n הוא לא מספר קרמייקל.

תהי $B = \left\{b \in \mathbb{Z}_n^* \mid b^{n-1} \equiv 1 \mod n\right\}$ אזי b לא ריקה כי $1 \in B$ ומכך ש-B סגורה לכפל מודולו n, אנו מקבלים כי B היא תת חבורה של $\mathbb{Z}_n^*$. מכך שכל "לא עד" a מקיים כי $a^{n-1} \equiv 1 \mod n$ אנו מקבלים כי B מכילה את קבוצת כל הלא עדים. בנוסף, מכך ש-$x \in \mathbb{Z}_n^* \backslash B$ אנו מקבלים כי $B \neq \mathbb{Z}_n^*$. כרצוי.

עתה, נניח כי לכל $x \in \mathbb{Z}_n^*$ מתקיים כי $x^{n-1} \equiv 1 \mod n$. כלומר n הוא מספר קרמייקל.

נניח בשלילה ש-n הוא חזקה של מספר ראשוני, שכן אחרת קיים p ראשוני כך ש-$n = p^e$ כאשר $e = v_p(n) > 1$. מההנחה ש-n אי זוגי מתקבל כי גם p ראשוני, ולכן קיים שורש פרימיטיבי g מודולו p^e, כלומר

$$\mathrm{ord}_{p^e}(g) = \varphi(p^e) = p^{e-1}(p-1)$$

מההנחה, $g^{n-1} \equiv 1 \mod n$ אבל מכאן $(p-1)p^{e-1} = \varphi(p^e) \mid n-1 = p^e - 1$ ולכן $p \mid p^e - 1$ כך $e > 1$, סתירה. לכן n לא יכול להיות חזקה של מספר ראשוני.

לכן קיימים $n_1, n_2 > 1$ כך ש-$n = n_1 \cdot n_2$ מספרים אי זוגיים זרים אחד לשני.

נגדיר t, u כך ש-$n - 1 = 2^t u$ עם u אי זוגי ו-$t \geq 1$. אזי פעולת ה-Witness מחשבת את הווקטור עבור איבר a:

$$X(a) = \left\langle a^u, a^{2u}, a^{2^2 u}, \ldots, a^{2^t u} \right\rangle (\mod n)$$

נגדיר זוג (v, j) להיות "מתקבל" כאשר $v^{2^j u} \equiv -1 \mod n$. בהכרח קיים אחד כזה שכן נוכל לבחור $v = n-1$ ו-$j = 0$. j מקסימלי עבורו קיים v כך שהתנאי מתקיים ונביט ב-

$$B = \left\{x \in \mathbb{Z}_n^* \mid x^{2^j u} \equiv \pm 1 \mod n\right\}$$

נבחין כי B היא סגורה לכפל ולכן היא תת חבורה של $\mathbb{Z}_n^*$. נוכיח כי B מכילה את כל הלא עדים.

יהי b איבר שאינו עד, נוכיח כי $b^{2^j u} \equiv \pm 1 \mod n$. נבחין כי הווקטור $X(b)$ מחשב את $b^{2^j u}$. אם $X(b)$ הוא ווקטור אחדות, סיימנו, אחרת מהיות b אינו עד, נוכל להביע על האינדקס המינימלי k עבורו $b^{2^k u} \equiv 1 \mod n$ אזי $b^{2^{k-1} u} \equiv -1 \mod n$ ממינימליות k. אבל ממקסימליות j מתקיים כי $j \geq k-1$ ולכן $b^{2^j u} \in \{-1, 1\}$. כלומר $b \in B$. עתה נוכיח כי $B \neq \mathbb{Z}_n^*$.

מהיות $v^{2^j u} \equiv -1 \mod n$ מתקיים גם כי $v^{2^j u} \equiv -1 \mod n_1$. ממשפט השאריות הסיני קיים w כך ש-

$$w \equiv v \mod n_1$$
$$w \equiv 1 \mod n_2$$

ולכן

$$w^{2^j u} \equiv v^{2^j u} \equiv -1 \mod n_1$$
$$w^{2^j u} \equiv 1 \mod n_2$$

נבחין כי $w^{2^j u} \not\equiv 1 \mod n$ וכן $w^{2^j u} \not\equiv -1 \mod n$ ולכן $w^{2^j u} \not\equiv \pm 1 \mod n$ ולכן $w \notin B$.

נוכיח כי $w \in \mathbb{Z}_n^*$ ונקבל כי $B \neq \mathbb{Z}_n^*$. מהיות $v \in \mathbb{Z}_n^*$ מתקיים כי $\gcd(n_1, v) = 1 = \gcd(n, v)$. מהיות $w \equiv v \mod n_1$ מתקיים כי $\gcd(w, n_1) = 1$ ובאותו האופן $\gcd(w, n_2) = 1$ ומהיות $\gcd(n_2, n_1) = 1$ מתקיים כי $\gcd(w, n) = 1$ ולכן $w \in \mathbb{Z}_n^*$. מכאן $w \in \mathbb{Z}_n^* \setminus B$ ולכן קיבלנו את הרצוי.

בשני המקרים, קיבלנו כי מספר העדים לפריקות הוא לכל הפחות $\frac{n-1}{2}$. כרצוי. ■ על כן אנו מקבלים את אלגוריתם מילר-רבין המלא הבא:

Algorithm 38 $\text{Miller} - \text{Rabin}(n, s)$

```
1 : for j = 1 to s :
2 :     a = Random(1, n − 1)
3 :     if Witness(a, n)
4 :         return "Composite"
5 : return "Prime"
```

משפט. לכל מספר אי זוגי $n > 2$ וטבעי s ההסתברות שאלגוריתם מילר רבין טועה היא לכל היותר 2^{-s}.

הוכחה: אם n ראשוני, בהכרח קיבלנו תשובה נכונה. אחרת, אם n פריק אזי צעדים $1-4$ באלגוריתם בעלי הסתברות של $\frac{1}{2}$ לגלות עד לפריקותו של n. מילר-רבין מבצע שגיאה אך ורק אם בחרנו איברים שאינם עדים בכל איטרציה, אבל זה קורה בהסתברות $\frac{1}{2^s}$. ■

32 תרגול 13 - מחלק משותף מקסימלי (gcd)

קלט $a, b \in \mathbb{N}$.

פלט $\gcd(a, b)$.

השערה. נעבור על כל המספרים הקטנים מ-a, b ונבדוק אם הם מחלקים את a, b. נחזיר את המספר המקסימלי שמצאנו.

בעיה. אנחנו רוצים אלגוריתם שיהיה פולינומיאלי באורך הקלט כלומר ב-$\log a$.

32.1 סיבוכיות פעולות אריתמטיות

יהיו $a, b \in \mathbb{N}$ שאורך הייצוג הבינארי שלהם חסום על ידי k.

- <u>חיבור וחיסור</u>: $\mathcal{O}\left(k\right)$.
- <u>כפל</u>: נוכל לעשות כפל ארוך ב-$\mathcal{O}\left(k^{2}\right)$.
- <u>חילוק</u>: $\mathcal{O}\left(k^{2}\right)$.
- <u>מודלו</u>: $\mathcal{O}\left(k^{2}\right)$ שכן $a\left(\mod b\right)=a-\left\lfloor \frac{a}{b}\right\rfloor \cdot b$.
- חישוב $a^{b-1}\mod b$: עולה $\mathcal{O}\left(k^{3}\right)$ כאשר אנו מפרקים את b לייצוג בינארי. שכן, נדרשות $\mathcal{O}\left(k\right)$ פעולות לחישוב $a^{2^{i}}$ לכל $0\leq i<k$ ועוד $\mathcal{O}\left(k\right)$ פעולות כפל לחישוב התוצאה ועוד $\mathcal{O}\left(k\right)$ פעולות של חישוב תוצאות הביניים מודולו b. כלומר, נדרשות פעולות החסומות כל אחת על ידי $\mathcal{O}\left(k^{2}\right)$.

הערה. החסמים הנ"ל הם עליונים ואינם הדוקים.

32.2 מציאת מחלק משותף מקסימלי

הגדרה. בהנתן $a,b\in\mathbb{N}\cup\{0\}$ כאשר $a\neq0$ נאמר ש-a מחלק את b ונסמן $a\mid b$ אם קיים $k\in\mathbb{N}\cup\{0\}$ כך ש-$b=a\cdot k$.

משפט. (החילוק עם שארית) בהנתן $a,b\in\mathbb{N}$ כך ש-$a\leq b$, קיים פירוק עם שארית יחיד מהצורה $k,r\in\mathbb{N}\cup\{0\}$ כך ש-$0\leq r<b$ ו-$a=k\cdot b+r$.

מסקנה. $k=\left\lfloor \frac{a}{b}\right\rfloor ,r=a\left(\mod b\right)$.

הגדרה. בהנתן $a,b\in\mathbb{N}\cup\{0\}$ כך ש-$a\leq b$ ו-$a\neq0$. <u>המחלק המשותף המקסימלי</u> של a,b המסומן ב-$\gcd\left(a,b\right)$ מוגדר להיות המספר המקסימלי $d\in\mathbb{N}$ כך ש-$d\mid a,d\mid b$.

הערות

1. $\gcd\left(0,0\right)$ לא מוגדר.
2. $\gcd\left(a,0\right)=a$ לכל $a\in\mathbb{N}$.

טענה. יהיו $a,b\in\mathbb{N}$ כך ש-$a\leq b$ אזי $\gcd\left(a,b\right)=\gcd\left(b,a\left(\mod b\right)\right)$.

הוכחה: נגדיר $d=\gcd\left(a,b\right)$ ו-$d'=\gcd\left(b,\left(a\mod b\right)\right)$. נראה כי $d\mid a\mod b,d'\mid a$, שכן אז $d\leq d'$ ולהפך.
נוכיח כי $d'\mid a$ נבחין כי קיימים $k_{1},k_{2}\in\mathbb{N}$ כך ש-$b=k_{1}d'$ ו-$a\left(\mod b\right)=k_{2}d'$ ולכן

$$a=\left\lfloor \frac{a}{b}\right\rfloor b+a\left(\mod b\right)=\left\lfloor \frac{a}{b}\right\rfloor \cdot k_{1}d'+k_{2}d'=d'\left(\left\lfloor \frac{a}{b}\right\rfloor \cdot k_{1}+k_{2}\right)$$

עתה נוכיח כי $d\mid a\left(\mod b\right)$. קיימים $k_{1},k_{2}\in\mathbb{N}$ כך ש-$a=k_{1}d,b=k_{2}d$ ולכן $a\left(\mod b\right)=a-\left\lfloor \frac{a}{b}\right\rfloor b=k_{1}d+\left\lfloor \frac{a}{b}\right\rfloor k_{2}d=d\left(k_{1}-\left\lfloor \frac{a}{b}\right\rfloor k_{2}\right)$. נבחין כי $k_{1}-\left\lfloor \frac{a}{b}\right\rfloor k_{2}$ חיובי שכן $a\left(\mod b\right),d>0$. אבל זה בעייתי כי יתכן כי $a\left(\mod b\right)=0$. אם זה המצב אז כל מספר מחלק אותו. אחרת, הוא לא אפס ואז ההסבר שנתנו תקף. ■

מסקנה. (אלגוריתם אוקלידס) נחזור על הפעולה $\gcd\left(a,b\right)=\gcd\left(b,a\left(\mod b\right)\right)$ ונחזיר את האיבר שאחריו נקבל אפס.

32.3 אלגוריתם אוקלידס המורחב

הגדרה. יהיו $a,b\in\mathbb{N}$ כך ש-$a\geq b$. נגדיר $S=\left\{ ax+by\mid{\scriptstyle \substack{x,y\in\mathbb{Z}\\ ax+by\geq1}}\right\}$.

טענה. $\gcd\left(a,b\right)=\min S$.

הוכחה: נגדיר $z=\min S$. $t=\gcd(a,b)$. נראה כי $z\le t$ וגם $z\ge t$.

נראה קודם ש-$z\le t$, בכך שנראה ש-$z\mid t$.

מהיות $z\in S$ קיימים $x,y\in\mathbb{Z}$ כך ש-$s=ax+by\ge 1$ קיימים $k_1,k_2\in\mathbb{N}$ כך ש-$a=k_1t, b=k_2t$ ולכן

$$s=ax+by=ak_1t+bk_2t=t\left(ak_1+bk_2\right)$$

שוב, $s>0$ וגם $t>0$ ולכן $ak_1+bk_2>0$.

נראה כי $t\le z$. נראה כי $z\mid a$ וגם $z\mid b$ ולכן $t\le z$ ממקסימליות t.

נרצה לפרק את a לפי z עם שארית כלומר $a=\left\lfloor\frac{a}{z}\right\rfloor z+a\left(\mod z\right)$. נוכיח קודם כל כי $z\le a$. נבחין כי $a=a\cdot 1+b\cdot 0\in S$ ולכן $z\le a$ ממינימליות z.

לכן אפשר לכתוב את $a=k\cdot z+r$. נוכיח כי $r=0$. נבחין כי $r=a-k\cdot z$ מהיות $z\in S$ קיימים $x,y\in\mathbb{Z}$ כך ש-$z=ax+by\ge 1$ ולכן

$$r=a-k\cdot(ax+by)=(1-kx)\,a+(-ky)\,b$$

אבל $0\le r<z$ ולכן ממינימליות z לא יתכן כי $r\in S$ ומכאן $r=0$. לכן $z\mid a$. בה"כ, נוכל לבצע אותו תהליך על b ולקבל כי $z\mid b$.
■

האלגוריתם

בעיה. בהנתן a,b נרצה למצוא s,t כך ש-$as+bt=\gcd(a,b)$.

ניזכר כי $\gcd(a,b)=\gcd\left(b,a\left(\mod b\right)\right)$ בנוסף, נזכור כי נוכל לרשום:

$$a=\left\lfloor\frac{a}{b}\right\rfloor b+r$$
$$bs+rt=\gcd(a,b)$$

ולכן נוכל לקבל:

$$r=a-\left\lfloor\frac{a}{b}\right\rfloor b\Rightarrow\left(bs+\left(a-\left\lfloor\frac{a}{b}\right\rfloor b\right)t\right)=at+\left(s-\left\lfloor\frac{a}{b}\right\rfloor t\right)b=\gcd(a,b)$$

כלומר נוכל להסיק את המקדמים לפי המקדמים מ-$b,a\left(\mod b\right)$. מכאן נסיק את ההליך הרקורסיבי הבא:

Algorithm 39 $\text{Extended}-\text{Euclid}\,(a,b)$

(1) : if $b=0$: return $(a,1,0)$ // $(\gcd(a,b),s,t)$
(2) : Let $(g',x',y')=\text{Extended}-\text{Euclid}\left(b,a\left(\mod b\right)\right)$
(3) : return $\left(g',y',x'-\left\lfloor\frac{a}{b}\right\rfloor\cdot y'\right)$

נכונות האלגוריתם

הוכחה: נוכיח את נכונות האלגוריתם באינדוקציה על מספר הקריאות הרקורסיביות k. כלומר שלכל זוג מספרים a,b עבורם האלגוריתם עוצר אחרי k איטרציות רקורסיביות, האלגוריתם מחזיר תשובה נכונה (g,x,y) כלומר $g=\gcd(a,b)=ax+by$.

<u>בסיס:</u> $k=0$: במקרה זה בהכרח $b=0$ ולכן $g=a=\gcd(a,b)=a\cdot 1+0\cdot b$.

שלב: נניח עבור k ונראה עבור $k+1$. מהנחת האינדוקציה $\text{Extened}-\text{Euclid}\left(b,a\left(\mod b\right)\right)$ מחזיר (g,x,y) כך ש-

$$g=\gcd\left(b,a\left(\mod b\right)\right)=bx+\left(a\left(\mod b\right)\right)y$$

מטענה קודמת, נובע כי $g=\gcd\left(a,b\right)=\gcd\left(b,a\left(\mod b\right)\right)$ ולכן, כמו שראינו קודם, קיימים x',y' מקיימים כי

$$ax'+bx'=g=\gcd\left(a,b\right)$$

כרצוי. ■

זמן ריצה

טענה. אחרי זוג קריאות רקורסיביות $\text{Extended}-\text{Euclid}$, שני הפרמטרים נחצים. כלומר, אם a,b הם הארגומנטים בקריאה כלשהי ו-a',b' אחרי 2 קריאות רקורסיביות אזי $b'\leq\frac{b}{2},a'\leq\frac{a}{2}$.

הוכחה: נבחין כי

$$\gcd\left(a,b\right)=\gcd\left(b,a\left(\mod b\right)\right)=\gcd\left(a\mod b\right),b\left(\mod\left(a\mod b\right)\right)$$

לכן מספיק להוכיח כי כל פעולת מודולו חוצה את המספר. ■

למה. יהיו $x,y\in\mathbb{N}$ כך ש-$x\geq y$ אזי $x\mod y\leq\frac{x}{2}$.

הוכחה: אם $y\leq\frac{x}{2}$ אזי $x\mod y<y\leq\frac{x}{2}$.
אחרת, אם $\frac{x}{2}<y\leq x$ אזי $\left\lfloor\frac{x}{y}\right\rfloor=1$ ולכן $x=y+x\left(\mod y\right)$. מכאן

$$x\mod y=x-y\leq x-\frac{x}{2}=\frac{x}{2}$$

■

מסקנה. מספר הקריאות הרקורסיבית הוא לכל היותר $2\log b$ ולכן לאחר $\mathcal{O}\left(\log b\right)$ האלגוריתם עוצר.
יתר על כן, אם $\log a,\log b\leq k$ אז נדרשות $\mathcal{O}\left(k^2\right)$ פעולות בכל קריאה ויש $\mathcal{O}\left(k\right)$ קריאות. לכן זמן הריצה הוא $\mathcal{O}\left(k^3\right)$.

32.4 מציאת איבר הפכי

בעיה. יהיו $a,n\in\mathbb{N}$ זרים. נרצה למצוא את ההפכי (היחיד) של a $\mod n$. כלומר נרצה לפתור את המשוואה $ax\equiv 1\mod n$.

אם a,n זרים, אזי $\gcd\left(a,n\right)=1$, נריץ את $\text{Exteded}-\text{Euclid}\left(a,n\right)$ ונקבל (g,x,y) כך ש-$ax+yn=1$ ולכן

$$ax=1-yn$$

נבחין כי $ax\equiv 1\mod n$ ולכן נוכל לחשב את שארית החלוקה של x ב-n ובזאת סיימנו.

33 תרגול 14

33.1 חלוקת הסדרה הממושקלת

קלט $w=\left(w_1,\dots,w_n\right)$ כאשר $w_i\in\mathbb{N}$, ומספר $m\in\mathbb{N}$.

פלט המטרה היא לחלק את הסדרה ל-m תתי סדרות רצופות כך שהמשקל של החלוקה מינימלי, משקל של חלוקה מוגדר להיות: נסכום כל תת סדרה ונחזיר את המקסימום.

דוגמה. $w=(1,8,7,10,40,15,30,2,1,20)$ עם $m=4$

חלק VIII

המבחן

34 חלק ראשון

בוחרים 2 מתוך 3. כל שאלה 30 נקודות. השאלות הן שאלות מוכרות, כלומר שאלות שראינו, ולכן הבדיקה שלהן נוקשה ודורשת ריגורוזיות רבה.

35 חלק שני

בוחרים 2 מתוך 3. כל שאלה 20 נקודות $\underbrace{5}_{\text{מוכר, למשל הגדרה}}+15$ כאשר 5 הנקודות אמורות להכווין אותנו לכיוון ה-15 האחרות. אלה שאלות חדשות, ולכן הבדיקה שלהן מתחשבת בטעויות קטנות.

הערה. בקרוב תמסר רשימת הנושאים שצריך לדעת לבחינה מתוך Crypto, RSA.