אלגוריתמים - 67504

חיים שחור - סיכומי תרגולים של צור לוריא

1 בפברואר 2012

בס"ד

תוכן עניינים

ה' מרחשוון תשע"ב (תרגול 1)

אדמיניסטרציה

צור לוריא: ה, 16-18 רוס 1-, חדר 36

אתר הקורס: www.cs.huji.ac.il/~algo

1 אלגוריתמים חמדניים

1.1 בעיית תחנות הדלק.

נוסעים ברכב לאורך קו ישר. המכל מספיק למספר מוגבל של קילומטרים. יש לנו מיקום של תחנות דלק לאורך הדרך, ואנו מעוניינים לעצור כמה שפחות פעמים.

קלט: N- מספר הקילומטרים שאפשר ליסוע עם מכל מלא, $a_1, \ldots, a_n$ המיקום של תחנות הדלק (מרחק מנקודת ההתחלה).

הנחות:

- $a_1 = 0$, ו-a_n הוא סוף המסלול.
- $\forall 1 \le i \le n-1 : a_{i+1} - a_i \le N$

פלט: תת-סדרה $b_1, \ldots, b_m$ של תחנות כך ש-$b_{i+1} - b_i \le N$, $b_1 = a_1$, $b_m = a_n$, m מינימלי.

פתרון: בכל תחנת דלק נבדוק האם יש לנו מספיק דלק כדי להמשיך לתחנה הבאה. אם אין מספיק דלק נעצור בתחנה.

סיבוכיות: $O(n)$, אנחנו עוברים לפי הסדר על כל התחנות.

משפט 1.1 הפתרון נותן פתרון אופטימלי.

הוכחה: חוקיות: צריך להראות שלכל i, $N \ge b_{i+1} - b_i$. נובע מההנחה שלכל i, $N \ge a_{i+1} - a_i$, ומהגדרת האלגוריתם.

אופטימליות: נראה באינדוקציה כי לכל $1 \le k \le m$, קיים פתרון אופטימלי שמסכים עם הפתרון שלנו ב-k הצעדים הראשונים.

בסיס האינדוקציה: $a_1 = b_1$ נובע מהגדרת הבעיה.

מעבר האינדוקציה: נסמן את הפתרון החמדן ב-$B = \{b_i \mid 1 \le i \le m\}$. נניח נכונות עבור $k-1$, לפי ה"א קיים פתרון אופטימלי $O = \{b_1, \ldots, b_{k-1}, c_k, \ldots c_l\}$. אם $c_k = b_k$ סיימנו את שלב האינדוקציה. אם $c_k \ne b_k$, נטען כי $b_k \ge c_k$ כי החמדן תמיד נוסע הכי רחוק שהוא יכול, בפרט מ-b_{k-1}. נסמן $O' = O \cup \{b_k\} \setminus \{c_k\}$ נותר להראות שהוא פתרון אופטימלי.

אופטימליות: O' הוא באורך l בדיוק כמו O ולכן גם O אופטימלי.

חוקיות: נשאר לבדוק את מה שהשתנה. אני יודע שעד b_k הפתרון חוקי כמו החמדן, ומ-c_{k+1} כמו O. נשאר לבדוק מ-b_k עד c_{k+1}. $c_{k+1} - b_k \le c_{k+1} - c_k \le N$ כי O חוקי. ∎

1.2 תזכורת - גרפים

הגדרה 1.2 גרף $G = (V, E)$, $E \subset \binom{V}{2}$.

הגדרה 1.3 שני קודקודים $u,v\in V$ שכנים אם $\{u,v\}\in E$.

הגדרה 1.4 מסלול מ־u ל־v זו סדרה של קודקודים $u=v_1,\dots,v_n=v$ כך ש־$\{v_i,v_{i+1}\}\in E$ לכל i.

הגדרה 1.5 מעגל: מסלול שמתחיל ומסתיים באותו קודקוד (באורך חיובי).

הגדרה 1.6 גרף נקרא קשיר אם יש מסלול בין כל שני קודקודים.

הגדרה 1.7 עץ הוא גרף קשיר וחסר מעגלים.
תכונות העץ:

1. $|E|=|V|-1$.
2. הוספת צלע לעץ תמיד יוצר מעגל יחיד.
3. תמיד יש מסלול יחיד בין כל שני קודקודים בעץ.

1.3 בעיית ה - MST עץ פורש מינימלי.

הגדרה 1.8 יהי $G=(V,E)$ גרף קשיר. עץ פורש ל־G זה גרף (V,T) כך ש־ (V,T) עץ, ו־$T\subseteq E$.

נתון גרף $G=(V,E)$, ופונקציית $w:E\to\mathbb{R}$ משקל. המשקל של עץ פורש T מוגדר להיות $w(T)=\sum_{e\in T}w(e)$.
מחפשים אלגוריתם שיחזיר עץ פורש במשקל מינימלי.
אלגוריתם חמדן לעפ"מ:

1. ממיינים את כל הצלעות לפי המשקל שלהם. $e_1,\dots,e_m$ כך ש־$w(e_i)\le w(e_{i+1})$.
2. $T=\phi$.
3. עוברים על הצלעות לפי הסדר, בכל צלע e, שואלים האם יש מעגל ב־$T\cup\{e\}$. אם יש מעגל ממשיכים הלאה, אם לא מוסיפים את e ל־T.

משפט 1.9 האלגוריתם נותן פתרון אופטימלי.

הוכחה: חוקיות:
נסמן את הפתרון החמדן ב־$T=\{t_1,\dots,t_{n-1}\}$. צריך להראות ש־T קשיר וחסר מעגלים.
T חסר מעגלים כי בכל פעם הוספנו צלע רק כשהיא לא יצרה מעגל ולכן בשום שלב לא נוצר מעגל ב־T.
נניח בשלילה ש־T לא קשיר. יש שתי קבוצות של קודקודים $C_1\cup C_2=V$ כך שלכל $u\in C_1,v\in C_2$, $\{u,v\}\notin T$.
אבל G קשיר ולכן יש ב־E צלע מ־C_1 ל־C_2. נקרא לו e. כשהאלגוריתם עבר על כל הצלעות הוא היה צריך לבחור בו, כי הוא מחבר שני רכיבי קשירות שונים ב־T.
אופטימליות:
נניח בשלילה ש־T לא אופטימלי. יהי S פתרון אופטימלי. נגדיר $k(S)=\max\{k\mid t_1,\dots,t_k\in S\}$ (לצורך העניין נגדיר $\max\phi=0$).
יהי S^* פתרון אופטימלי שממקסם את $k(S)$. נסמן $k^*=k(S^*)$. מההנחה בשלילה $k^*<n-1$.
$S^*=\{t_1,\dots,t_{k^*},r_{k^*+1},\dots,r_{n-1}\}$. נרצה לבנות פתרון אופטימלי שמכיל את $\{t_1,\dots,t_{k^*+1}\}$ וכך נקבל סתירה

(למקסום של S^*). נתבונן ב־$S^*\cup\{t_{k^*+1}\}$. בגרף זה יש מעגל יחיד C. קיימת צלע ב־C שלא שייך ל־T. נקרא לו e.
נגדיר $S'=S\cup\{t_{k^*+1}\}\setminus\{e\}$. עלינו להראות כי S' הוא פתרון אופטימלי וסיימנו.
חוקיות: ב־$S^*\cup\{t_{k^*+1}\}$ יש מעגל יחיד C, ו־$e\in C$, ולכן C לא נמצא ב־S'. S' חסר מעגלים בעל $n+1$ צלעות לכן S' עץ פורש.
אופטימליות: $w\left(S'\right)=w\left(S^*\right)-w\left(e\right)+w\left(t_{k-1}\right)$, נראה כי $w\left(e\right)\geq w\left(t_{k^*+1}\right)$, ואז $w\left(S'\right)\leq w\left(S^*\right)$, כלומר S' מינימלי. האלגוריתם עבר על הצלעות לפי סדר עולה של הצלעות. אנו יודעים כי $t_1,\ldots,t_{k^*}\in S$, ולכן אם נוסיף להם את e, לא ייסגר מעגל. אם האלגוריתם פגש את e לפני t_{k^*+1}, הוא היה צריך להוסיף אותו לT, ואם הוא לא עשה כן, סימן ש־$w\left(e\right)\geq w\left(t_{k^*+1}\right)$. ∎

י"ב מרחשוון תשע"ב (תרגול 2)

1.4 בעיית התרמיל השברית (Knapsack)

גנב נכנס לחנות, ולכל חפץ יש משקל, כאשר לגנב יש מגבלת משקל.
קלט: W המשקל המקסימלי שהגנב יכול לסחוב. ורשימה של $\{(v_i,w_i)\}_{i=1}^n$.
פלט: רשימה $(x_i)_{i=1}^n\in\{0,1\}^n$ כך ש־

1. $\sum_{i=1}^n x_iw_i\leq W$.
2. $\sum_{i=1}^n x_iv_i$ מקסימלי.

הבעיה הזו מסובכת (NPC), אבל נסתכל על בעיה קצת שונה ־ בעיית KS השברית.
קלט: אותו הדבר.
פלט: רשימה $(x_i)_{i=1}^n\in[0,1]^n$ תחת אותם תנאים (ניתן לקחת גם חצאי חפצים).
פתרון:

- נגדיר לכל חפץ ערך סגולי (לפי המשקל) $r_i=\dfrac{v_i}{w_i}$
- ממיינים את החפצים לפי r_i, $(r_1\geq r_2\geq\ldots\geq r_n)$.
- עוברים על החפצים, ובשלב הk בודקים:
  - אם $W\geq\sum_{i=1}^k w_i$, אז $x_k=1$.
  - בשלב הראשון t שבו $\sum_{i=1}^t w_i<W$, ניקח $x_t=\dfrac{W-\sum_{i=1}^{k-1}w_i}{w_t}$.
  - לכל $k>t$ ניקח $x_k=0$.

דוגמא: נניח $W=10$, $\begin{pmatrix}v_1=5\\w_1=1\end{pmatrix},\begin{pmatrix}v_2=10\\w_2=10\end{pmatrix}$. בבעיה השברית הוא ייקח את כל $x_1=1$, ו־$x_2=0.9$. בסה"כ המשקל יהיה $5+0.9\cdot10=14$. **הוכחה:** חוקיות:

1. $0\leq x\leq1$: מספיק לבדוק עבור x_t. $0=\dfrac{W-W}{W_t}\leq\dfrac{W-\sum_{i=1}^{t-1}w_i}{w_t}=x_t<\dfrac{\sum_{i=1}^{t}w_i-\sum_{i=1}^{t-1}w_i}{w_t}=\dfrac{w_t}{w_t}=1$.

2. צ"ל $\sum_{i=1}^{n} x_i w_i \le W$. אנחנו נראה שוויון (בהנחה שסכום המשקלות הכולל גדול מ־W, אחרת ניתן פשוט לקחת את כל החפצים).

$$\sum_{i=1}^{n} x_i w_i = \sum_{i=1}^{t-1} w_i + x_t w_t = \sum_{i=1}^{t-1} w_i + \frac{W - \sum_{i=1}^{t-1}}{w_t} w_t = W$$

<u>אופטימליות</u>: צ"ל שלכל פתורן חוקי $y_1, \ldots, y_n$ מתקיים $\sum_{i=1}^{n} y_i v_i \le \sum_{i=1}^{n} x_i v_i$. יהי (y_i) פתרון חוקי אחר. נראה כי $\sum (x_i - y_i) v_i \ge 0$:

$$\begin{aligned} \sum_{i=1}^{n} (x_i - y_i) v_i &= \sum (x_i - y_i) w_i r_i = \sum_{i=1}^{t-1} (x_i - y_i) w_i r_i + (x_t - y_t) w_t r_t + \sum_{i=t+1}^{n} (x_i - y_i) w_i r_i \ge \\ &\ge \sum_{i=1}^{t-1} (1 - y_i) w_i r_t + (x_t - y_t) w_t r_t + \sum_{i=t+1}^{n} (-y_i) w_i r_t = r_t \sum_{i=1}^{n} (x_i - y_i) w_i = \\ &= r_t \left(\sum_{i=1}^{n} x_i w_i - \sum_{i=1}^{n} y_i w_i \right) = r_t \left(W - \sum_{i=1}^{n} y_i w_i \right) \ge 0 \end{aligned}$$

כי $W \ge \sum y_i w_i$. ■

1.5 מטרואידים

הגדרה 1.10 מטרואיד הוא $M = (S, I)$. S קבוצה סופית. $I \subseteq P(S)$ כך ש־

1. תורשתיות: $B \subset A \in I \Rightarrow B \in I$.
2. החלפה: אם $A, B \in I, |A| \ge |B| \Rightarrow \exists a \in A \backslash B \text{ s.t. } \{a\} \cup B \in I$.

באופן כללי זו בעיה קומבינטורית שבה I הוא אוסף הפתרונות החוקיים, אבל יש תנאים נוספים.

1.5.1 דוגמאות

מטרואיד אוניפורמי: $S = \{1, \ldots, n\}, I = \{A \subseteq S \mid |A| \le k\}$

טענה 1.11 זה מטרואיד.

הוכחה: <u>תורשתיות</u>: אם $A \in I$ אז $|A| \le k$, ואם $B \subset A$ אז $|B| \le |A|$ ולכן גם $|B| \le k$ כלומר $B \in I$.
<u>החלפה</u>: נניח $|B| < |A| \le k$, $A, B \in I$ אזי $|B| \le k - 1$, ולכן לכל $a \in A \backslash B$, יתקיים $|\{a\} \cup B| = |B| + 1 \le k$ כלומר $\{a\} \cup B \in I$. ■

המטרואיד המטריציוני: תהי $T \in M_{n,m}$ מטריצה, ויהיו $t_1, \ldots, t_n$ שורות המטריצה. $S = \{t_1, \ldots, t_n\}$. $I = \{A \subseteq S \mid *\}$: * ־ הוקטורים ב־$A$ הם בת"ל.

טענה 1.12 זהו מטרואיד.

הוכחה: <u>תורשתיות</u>: נובע מאלגברה לינארית.
<u>החלפה</u>: נניח $A, B \in I, |A| > |B|$. צריך להראות שקיים $v \in A \backslash B$ כך ש־$v \cup \{B\}$ בת"ל. v ב"ת בוקטורים ב־B אס"ם $v \notin Span\left(B\right)$. אבל $\dim Span\left(A\right) = |A| > |B| = \dim Span\left(B\right)$, לכן לא כל הוקטורים ב־$A$ יכולים להיות ב־$Span\left(B\right)$ אחרת A לא היתה בת"ל לכן יש $v \in A \not\ni Span\left(B\right)$. ■

המטרואיד הגרפי: יהי $G = (V, E)$ גרף קשיר. נגדיר $S = E$, $I = \{A \subseteq S \mid A \text{ don't contain circle}\}$ (איבר ב־I הוא יער). נשים לב כי בגרף (V, A) יש $|V| - |A|$ רכיבי קשירות.

טענה 1.13 זהו מטרואיד.

הוכחה: <u>תורשתיות</u>: אם A חסר מעגלים, אז ברור שגם כל תת־קבוצה של A חסרת מעגלים.
<u>החלפה</u>: נניח $A, B \in I$. $|A| > |B|$. ב־B יש יותר רכיבי קשירות מאשר ל־A. לכן לא כל רכיב קשירות של A מוכל ברכיב קשירות של B. קיים ב־A רכיב קשירות L שאינו מוכל ברכיב קשירות. לכן קיים רכיב קשירות C ב־B כך ש־ $L \neq C \cap L \neq \phi$ ו־$L \backslash C \neq \phi$.לכן יש קודקוד $u \in C \cap L, v \in C \backslash L$. L קשיר, ולכן יש מסלול מ־u ל־v ב־L. במסלול הזה יש צלע e מ־C ל־$L \backslash C$ ולכן $B \cup \{e\}$ חסר מעגלים כלומר $B \cup \{e\} \in I$. ■

1.6 האלגוריתם החמדן

נניח שיש לנו פונקציה $w : S \to \mathbb{R}^+$. רוצים למצוא $A \in I$ כך ש־$w\left(A\right) = \sum_{a \in A} w\left(a\right)$ מקסימלי.
אלגוריתם חמדן:

- נמיין את איברי S כך ש־$w\left(a_1\right) \geq w\left(a_2\right) \geq \ldots \geq w\left(a_n\right)$.
- נאתחל $A := \phi$.
- נעבור על האיברים לפי הסדר:
  - בכל שלב נבדוק האם $A \cup \{a_i\} \in I$.
    - אם כן נצרף את a_i ל־A.

מבחינת זמן ריצה, יש לנו את המיון של $\theta\left(n \log n\right)$, אבל גם את הבדיקה של $A \cup \{a_i\}$.
האלגוריתם החמדן על מטרואיד הגרף:

1. ממיינים את הצלעות.
2. מאתחלים $T = \phi$.
3. בשלב ה־k, אם $T \cup \{e_k\}$ חסר מעגלים, מוסיפים את e_k ל־T.

איך מממשים את השלב האמצעי? מהו זמן הריצה לבדיקת האם צלע סוגרת מעגל?
דרך אחת: להתחיל מ־u, להריץ BFS ולבדוק אם מגיעים ל־v. מאחר ומספר הצלעות הוא לכל היותר $|V|$, הסיבוכיות היא $O\left(|V|\right)$, ואם חוזרים $|E|$ פעמים על הבדיקה נקבל $O\left(|E|\,|V|\right)$.
דרך שניה: להחזיק לכל קודקוד לאיזה רכיב קשירות הוא נמצא. אנחנו יורדים ל־$O\left(|V|^2\right)$.

2 רשתות זרימה

י"ט מרחשוון תשע"ב (תרגול 3)

2.1 חזרה על הגדרות

הגדרה 2.1 רשת זרימה מורכבת מגרף מכוון $G=\langle V,E\rangle$ עם 2 קודקודים מיוחדים $s,t\in V$. וקיבולת $c:E\to\mathbb{R}_+$.

הגדרה 2.2 זרימה בר"ז $f:E\to\mathbb{R}^+$ כך ש-

1. לכל $e\in E$, $f(e)<c(e)$.
2. שימור החומר: $\forall s,t\neq v\in V:\ \sum_{e\in in(v)}f(e)=\sum_{e\in out(v)}f(e)$.

הגדרה 2.3 הגודל של זרימה f הוא $\sum_{e\in out(s)}f(e)\left(-\sum_{e\in in(s)}f(e)\right)$ (אם נניח שלא נכנסים למקור אפשר להשמיט את החלק הימני).

הגדרה 2.4 חתך הוא חלוקה של V לשתי קבוצות לא ריקות $S,V\backslash S$ כך ש-$s\in S,t\in V\backslash S$.

הגדרה 2.5 קיבול של חתך יוגדר ע"י $C(S,V\backslash S)=\sum_{e\in S\times V\backslash S}c(e)$. הקיבול של כל חתך חוסם כל זרימה.

הגדרה 2.6 חתך מינימלי זהו חתך שהקיבול שלו מינימלי.

הגדרה 2.7 בהינתן רשת זרימה G,s,t,c וזרימה חוקית f. נגדיר את הרשת השיורית G_f ע"י:

- קבוצת הקודקודים היא V. המקור הוא s, היעד הוא t (אותו דבר).
- הצלעות: נגדיר $E'_f=\{(u,v)\mid(u,v)\in E\vee(v,u)\in E\}$. ניקח ל-$E_f=\left\{e\in E'_f\mid c_f(e)>0\right\}$.
- הקיבולת: $c_f(u,v)=c(u,v)-f(u,v)+f(v,u)$ (כאשר $e\notin E\Rightarrow c(e)=f(e)=0$).

2.2 תכונות

1. לכל חתך $S,V\backslash S$, סך הזרימה בחתך שווה לגודל הזרימה.
2. לכל חתך ולכל זרימה $C(S,V\backslash S)\geq|f|$.
3. אם $c(e)\in\mathbb{N}$ לכל $e\in E$, אז יש זרימה מקסימלית f כך ש-$f(e)\in\mathbb{N}$ לכל e, וזרימה כזו נקראת זרימה שלמה.

משפט 2.8 החתך בעל הקיבולת המינימלית, מכיל קיבולת זהה לשטף הזרימה המקסימלי. $\mathrm{MINCUT}=\mathrm{MAXFlow}$.

2.2.1 שיטת FF.

תהי G,s,t,c רשת זרימה. רוצים זרימה f בגודל מקסימלי.
נאתחל $f\equiv 0$.
הרעיון: בכל שלב ננסה להגדיל את $|f|$, כך שנשמור על החוקיות של f, וכשלא נוכל יותר לשפר את f הזרימה תהיה מקסימלית.
בכל שלב יש לנו את G,s,t,c ואת f הנוכחית. נגדיר את G_f ונחפש מסלול ברשת השיורית מ־s ל־t (שכל צלעותיו בעלות קיבולת חיובית).
נעביר במסלול זרימה ככל האפשר (שווה לקיבול המינימלי של צלע במסלול) ונוסיף את המסלול ל־f (צריך להגדיר חיבור זרימות).

2.3 זיווג מקסימלי בגרף.

יהי $G=\langle L,R,E\rangle$ גרף דו־צדדי. כלומר כל צלע $e\in E$ מחברת קודקוד של L עם קודקוד של R. זיווג בגרף זה $M\subseteq E$ כך ש־$\forall v\in L\cup R, \deg v\leq 1$. הגודל של זיווג M זה מספר הצלעות בו $|M|$.
הבעיה: בהינתן עץ דו־צדדי, נרצה למצוא זיווג בגודל מקסימלי.
אם $|L|=|R|=n$ זיווג מושלם הוא זיווג עם n צלעות.
אלגוריתם למציאת זיווג מקסימלי בגרף דו"צ (עם רשת זרימה):
נתון $G=\langle L\cup R,E\rangle$. נגדיר רשת זרימה באופן הבא:

- $V=L\cup R\cup\{s,t\}$.
- $E'=\{(l,r)\mid \{l,r\}\in E\}\cup\{(s,l)\mid l\in L\}\cup\{(r,t)\mid r\in R\}$.
- $\forall e\in E': c(e)=1$

כעת נמצא זרימה מקסימלית f שלמה בעזרת אחד האלגוריתמים שנלמדו בכתה. ניקח $M=\{\{l,r\}\in E\mid f(l,r)=1\}$.

טענה 2.9 M חוקי ואופטימלי.

הוכחה: חוקיות: צריך להוכיח כי כל קודקוד נמצא בלכל היותר צלע אחת של M. לפי שימור החומר, לקודקוד ב־L נכנס לכל היותר 1, ולכן צריך לצאת לכל היותר אחד בשל שימור הזרימה. זרימה זו לא מתחלקת לשתי קשתות כי f זרימה שלמה. באופן דומה לגבי R יוצא לכל היותר 1.
אופטימליות: נניח בשלילה שיש זיווג M' כך ש־$|M'|>|M|$. נשתמש ב־M' כדי להגדיר זרימה f' כך ש־$|f'|\geq|f|$ וזו סתירה:
נשים לב כי $|M|=|f|$ כי סך הזרימה שזורמת בחתך $|M|=(\{s\}\cup L,\{t\}\cup R)=|f|$. נגדיר את f' ע"י:

- $f'(s,l)=1$ אם"ם l מזווג ב־M'.
- $f'(l,r)=1 \Leftrightarrow \{l,r\}\in M$
- $f'(r,t)=1$ אם"ם r מזווג ב־M'.

f' חוקי כי:

1. אילוצי קיבולת: f' מזרים זרימה רק על צלעות של הרשת, והוא מזרים 0 או 1 על כל צלע.

2. חוק שימור החומר: עבור $l \in L$, אם l מזווג ב־M', אזי $f(s,l) = 1$, M' זיווג, ולכן l משתתף בדיוק בצלע אחת $\{l,r\}$ ולכן $f(l,r) = 1$ עבור צלע אחת בדיוק, ולכן חוק שימור החומר מתקיים.

$|M'| = |f'|$ מאותו שיקול שהראה $|M| = |f|$. קיבלנו $|f'| > |f|$ בסתירה למקסימליות f.

■

משפט 2.10 החתונה של Hall: $|R| = |L| = n$, יש ב־G זיווג מושלם אם לכל $X \subseteq L$ מתקיים $|N(X)| \geq |X|$.
הכללה: הגודל של זיווג מקסימלי בגרף דו"צ זה $n - \max_{X \subseteq L} \{|X| - |N(X)|\}$.

הוכחה: (רעיון) נראה שהחתך המינימלי הוא בעל קיבול של $n - \max_{X \subseteq L} \{|X| - |N(X)|\}$ ■

2.4 מציאת חתך מינימלי

נתונה רשת זרימה. רוצים למצוא חתך מינימלי.

1. מוצאים זרימה מקסימלית f.
2. מחשבים את G_f.
3. נסמן $S = \{v \in V \mid v \text{ accessible from } s\}$.

טענה 2.11 $(S, V \backslash S)$ חתך מינימלי.

הוכחה: נראה $C(S, V \backslash S) = |f|$.

1. לכל צלע $e \in S \times V \backslash S$, $f(e) = c(e)$, אחרת אם $f(e) < c(e)$ אז e קיימת ברשת השיורית (עם קיבול $c(e) - f(e)$), ולכן יש מסלול מ־S לקודקוד ב־$V \backslash S$.
2. לכל צלע $e \in V \backslash S \times S$, $f(e) = 0$. אחרת ב־G_f היתה צלע מ־S ל־t עם קיבול $f(e)$ חיובי.

$$|f| = \sum_{e \in S \times V \backslash S} f(e) - \sum_{e \in V \backslash S \times S} f(e) = \sum_{e \in S \times V \backslash S} c(e) = C(S, V \backslash S)$$

■

2.5 בעיית ניקוי האולם

כ"ו מרחשוון תשע"ב (תרגול 4)
נתונים:

- סטודנטים $s_1, \ldots, s_n$.
- ימים $d_1, \ldots, d_m$.
- לכל סטודנט S_i, קבוצה $D_i \subset D = \{d_1, \ldots, d_m\}$ של ימים בהם הוא מגיע.

- לכל יום d_i, קבוצת סטודנטים $S_i \subseteq S = \{s_1, \ldots, s_n\}$ של סטודנטים שבאים ביום זה.
- $\forall i: |S_i| > 0$.

מטרה: נסמן $P_i = \sum_{d_j \in D_i} \frac{1}{|S_j|}$. רוצים לבחור סטודט לכל יום שינקה את האולם, כך שמספר הפעמים שהסטודנט ה־i יבחר לא יעלה על $P'_i = \lceil P_i \rceil$.

פתרון בעזרת רשת זרימה: נגדיר רשת זרימה ע"י: $G = (V, E)$, כאשר $V = \{s, t\} \cup S \cup D$.
$E = \{(s, s_i) \mid 1 \le i \le n\} \cup \{(d_j, t) \mid 1 \le j \le m\} \cup \{(s_i, d_j) \mid d_j \in D_i\}$. קיבול $c(e) = \begin{cases} P'_i & e = (s, s_i) \\ 1 & o.w. \end{cases}$.

הערה: מספר הצלעות שנכנסות ל־d_j הוא $|S_j|$.

תהי f זרימה שלמה ברשת בגודל m. נוכל לפתור את הבעיה אם נותנים כל סטודנט לימים שאליהם זורמת ממנו זרימה.

תזכורת: אם כל הקיבולות שלמות, תמיד קיימת זרימה מקסימלית שלמה.

טענה 2.12 תמיד יש זרימה בגודל m ברשת.

הוכחה: נגדיר f באופן הבא: $f(s, s_i) = P_i$. $f(s_i, d_j) = \frac{1}{|S_j|}$. $f(d_j, t) = 1$. (זוהי זרימה לא־שלמה, אבל אפשר למצוא זרימה שלמה בקיבול כזה).

f חוקית כי:

1. הזרימה תמיד קטנה מהקיבול. $P_i \le P'_i$, וכל שאר הזרימות ≤ 1.
2. חוק שימור החומר: צריך להוכיח שלכל $v \in S \cup D$ הזרימה הנכנסת שווה לזרימה היוצאת. עבור $s_i \in S$ נכנס P_i, ויוצא $\sum_{d_j \in D_i} \frac{1}{|S_j|} = P_i$. עבור $d_j \in D$, יוצא 1, ונכנס $\sum_{s_i \in S_j} \frac{1}{|S_j|} = 1$.

מסקנה 2.13 יש זרימה בגודל m. היא זרימה מקסימלית, ולכן יש זרימה שלמה בגודל m.

■

2.6 בעיית השחקנים והמשקיעים

נתונים:

- אוסף שחקנים $A = \{a_i \mid 1 \le i \le n\}$.
- השחקן a_i רוצה משכורת $s_i \in \mathbb{N}_+$.
- אוסף משקיעים $I = \{I_j \mid 1 \le j \le k\}$.
- המשקיע I_j תורם r_j שקלים, אבל מוכן לתרום רק אם השחקנים האהובים עליו $F_j \subseteq A$ משחקים.

רוצים למקסם את הרווח $\sum r_j - \sum s_i$.

הערה: באופן שקול רוצים למצוא $X \subseteq I$ שממקסמת את $\sum_{I_j \in X} r_j - \sum_{a_i \in \bigcup_{I_j \in X} F_j} s_i$.

פתרון: נגדיר רשת זרימה ע"י $V = A \cup I \cup \{s, t\}$. נגדיר $c(s, I_j) = r_j$, $c(a_i, t) = s_i$, ו־$c(I_j, a_i) = \infty$.

האלגוריתם: נמצא חתך מינימלי $S, V \backslash S$ בגרף. ונחזיר $X = S \cap I$.

טענה 2.14 האלגוריתם ממקסם את הרווח.

הוכחה: צ"ל X ממקסם את $\sum_{I_j\in X} r_j - \sum_{a_i\in\bigcup_{I_j\in X}F_j} s_i$. נראה שהקיבול של חתך מינימלי בגרף שווה ל־

$$\sum_{I_j\in I} r_j - \max_{X\subseteq I}\left(\sum_{I_j\in X} r_j - \sum_{a_i\in\bigcup_{I_j\in X}F_j} s_i\right)$$

יהי S חתך. נסמן $X=S\cap I$ ו־$Y=A\cap S$, ונניח $C(S,V\backslash S)<\infty$ (קיים חתך סופי, ולכן ניקח את כל הסופיים כמועמדים לחתך מינימלי). מהנחה זו, $S\supseteq\bigcup_{I_j\in X}F_j$, אחרת הקיבול יהיה ∞. לכן יש לנו שני סוגים של צלעות שיוצאות מהחתך. מהמקור ל־$I\backslash X$, ומ־Y לבור. כלומר

$$C(S,V\backslash S)=\sum_{I\in I\backslash X} r_i+\sum_{a_i\in Y} S_i\geq\sum_{I_j\in I} r_j-\sum_{I_j\in X} r_j+\sum_{a_i\in\bigcup_{I_j\in X}F_j} s_i$$

כאשר שוויון יתקבל אם $Y=\bigcup F_j$. יהי X שממקסם את הרווח. נתבונן בחתך $S=\{s\}\cup X\cup\bigcup_{I_j\in X}F_j$, אזי הקיבול שלו הוא $\sum_{I_j\in I} r_j-\max\left(\sum_{I_j\in X} r_j-\sum_{a_i\in\bigcup_{I_j\in X}F_j} s_i\right)$. ■

2.7 מסלולים זרים בגרף

יהי G גרף מכוון. יהיו u,v קודקודים. נאמר ששני מסלולים $u\to v$ זרים בצלעות אם אין להם צלעות משותפות. נאמר ששני מסלולים זרים בקודקודים אם אין להם קודקודים פנימיים משותפים (זרות בקודקדים $\Leftarrow$ זרות בצלעות).

שאלה: בהינתן G,u,v, מצא k מסלולים זרים בצלעות $u\to v$, או החזר שאין כאלו.

נגדיר רשת זרימה ע"י $s=u,t=v$, וקיבול 1 לכל הצלעות. נבדוק האם יש זרימה שלמה בשטף k.

מה לגבי זרות בקודקודים? נפצל כל קודקוד לשנים, אחד יקבל את כל הצלעות הנכנסות, ואחד יוציא את כל הצלעות היוצאות, עם צלע ביניהם בקיבול 1.

2.8 חזרה על אלגוריתם Dinic

ד' כסלו תשע"ב (תרגול 5)

הגדרה 2.15 רשת שכבות - בהינתן רשת G,c,s,t נגדיר $L(G)$ להיות הרשת שמתקבלת מ־G אם זורקים את כל הצלעות והקודקודים שלא משתתפים באף מסלול קצר ביותר $s\to t$.

הגדרה 2.16 זרימה חוסמת - זרימה f כך שעל כל מסלול $s\to t$ ברשת יש צלע רוויה.

האלגוריתם:

- מאתחלים $f\equiv 0$.
- חוזרים עד שהגענו לזרימה מקסימלית
  - מוצאים את $L(G_f)$.
  - מוצאים זרימה חוסמת g ב־$L(G_f)$.

- מוסיפים את g ל־f.

האלגוריתם לא מושלם, עד שלא נגדיר אלגוריתם למציאת $L\left(G\right)$, ולמציאת זרימה חוסמת.
מציאת $L\left(G\right)$:

- מריצים BFS על G, ומסמנים כל קודקוד עם המרחק שלו מ־s.
- מעיפים צלעות $\left(x,y\right)$ אם $l\left(x\right)+1>l\left(y\right)$.
- עוברים על כל הקודקודים (מהשכבות הרחוקות למקור), ומורידים קודקוד אם אין לו צלעות יוצאות.

מציאת זרימה חוסמת בגרף שכבות:

- מאתחלים $g\equiv0$.
- כל עוד ניתן
  - מוצאים מסלול $p:s\to t$ בצורה חמדנית.
  - מזרימים ב־p זרימה של $c\left(p\right)=\min_{e\in p}c\left(e\right)$.
  - מוסיפים את הזרימה ל־g.
  - מעדכנים את הקיבולות בהתאם ברשת.
  - מעדכנים את הרשת: אם לקודקוד אין צלעות נכנסות (בסדר עולה) או יוצאות (בסדר יורד) מורידים אותם.

3 תכנות דינאמי

3.1 בעיית תת־המחרוזת המשותפת הארוכה ביותר LCS

קלט: 2 מחרוזות $X=x_{1},\ldots,x_{n}$ ו־$Y=y_{1},\ldots,y_{m}$.
מחפשים את תת־המחרוזת המשותפת (אפשר עם דילוגים, אבל לשמור על הסדר) באורך מירבי.
לדוגמא: $X=abccba,Y=bacbc$, אז abc תת־מחרות משותפת.
הרעיון הוא לבנות פתרון אופטימלי לבעיה הגדולה, בעזרת תתי בעיות קטנות יותר.
נגדיר $OPT\left(i,j\right)$ להיות האורך של תת־המחרוזת הארוכה ביותר של $X_{i}=x_{1},\ldots,x_{i}$ ו־$Y_{j}=y_{1},\ldots,y_{j}$. אנו מחפשים את $OPT\left(n,m\right)$. אם $x_{n}\neq y_{m}$, לא ניתן לקחת את שתי האותיות האחרונות. לכן $OPT\left(n,m\right)=\max\left\{ OPT\left(n,m-1\right),OPT\left(n-1,m\right)\right\}$. אם $x_{n}=y_{m}$, אז $OPT\left(n,m\right)=OPT\left(n-1,m-1\right)$.

טענה 3.1 לכל $0\leq i\leq n,\,0\leq j\leq m$,
$$OPT\left(i,j\right)=\begin{cases}0 & i\cdot j=0\\ \max\left\{ OPT\left(i,j-1\right),OPT\left(i-1,j\right)\right\} & x_{i}\neq y_{j}\\ OPT\left(i-1,j-1\right)+1 & x_{i}=y_{j}\end{cases}.$$

הוכחה: נראה שוויון בכל אחד מהמקרים:

1. אם $i=0$ אז $X_{0}=\phi$, וכל תת־מחרוזת של X_{0} היא ϕ, לכן $LSC=0,OPT\left(0,j\right)=0$. באופן דומה עבור $j=0$.

2. אם $x_i \neq y_j$.

(א) נראה $OPT(i,j) \leq \max\{OPT(i,j-1), OPT(i-1,j)\}$: $x_i \neq y_j$, לכן התת"מ לא יכולה לקחת גם את x_i וגם את y_j. נניח בה"כ שהיא לא לוקחת את y_j, ולכן היא גם תת־מחרוזת של X_i, Y_{j-1}, ולכן אורכה קטן או שווה ל־$OPT(i,j-1)$. באופן דומה אם התת"מ לא לוקחת את x_i, אז אורכה קטן או שווה ל־$OPT(i-1,j)$.

(ב) נראה כי $OPT(i,j) \geq \max\{OPT(i-1,j), OPT(i,j-1)\}$. כל תת"מ של X_{i-1}, Y_j היא גם תת"מ של X_i, Y_j, ולכן ניתן לבנות תת"מ של X_i, Y_j באורך של $OPT(i-1,j)$, ובאופן דומה באורך של $OPT(i,j-1)$.

3. אם $x_i = y_j$

(א) אם z תת"מ מקסימלית של X_{i-1}, Y_{j-1} אז z, x_i היא תת"מ של X_i, Y_j באורך $OPT(i-1,j-1)+1$, כלומר $OPT(i,j) \geq OPT(i-1,j-1)$.

(ב) אם $z = z_1, \ldots, z_k$ תת"מ משותפת מקסימלית ל־X_i, Y_j, אז $z_1, \ldots, z_{k-1}$ תת"מ חוקית ל־X_{i-1}, Y_{j-1}, כלומר $OPT(i-1,j-1) \geq OPT(i,j)-1$, ובסה"כ $OPT(i,j) = OPT(i-1,j-1)+1$.

■

האלגוריתם:

- נאתחל $f(0,j) = f(i,0) = 0$, לכל $0 \leq i \leq n,\ 0 \leq j \leq m$.
- נמלא טבלה בגודל $(n+1)\times(m+1)$, לפי הסדר של השורות, ונחשב את $f(i,j)$ בעזרת הנוסחא הרקורסיבית. נחזיר את $f(n,m)$.

דוגמא עבור המחרוזות $X = acab, Y = bac$, נאתחל טבלה

i/j	0	1	2	3
4	0	1	1	2
3	0	0	1	2
2	0	0	1	2
1	0	0	1	1
0	0	0	0	0

, כאשר מהטבלה ניתן להסיק את המחרוזת עצמה, אם שומרים על "מאיפה הגענו".

טענה 3.2 נראה באינדוקציה כי $f(i,j) = OPT(i,j)$

הוכחה: בסיס האינדוקציה: $i \cdot j = 0$, אז $OPT(i,j) = 0 = f(i,j)$.
צעד האינדוקציה: נניח נכונות עבור $i+j$ קטן יותר, ובפרט עבור $i-1, j;\ i, j-1;\ i-1, j-1$. אזי

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

כנדרש. ■

סיכום: יש לנו בעיה, מפצלים אותה לתתי־בעיות קטנות יותר, וקשר רקורסיבי ביניהם. פותרים את כל תתי הבעיות לפי הסדר של הטבלה, עד שמחשבים את תת־הבעיה הכי גדולה, שהיא הטבלה המקורית.
זמן הריצה: מספר תתי־הבעיות, כפול הזמן שלוקח לפתור אחת מהן. במקרה שלנו $O(nm)$.

3.2 בעיית כפל המטריצות

י"א כסלו תשע"ב (תרגול 6)

כשמכפילים $A\left[n\times m\right]$, $B\left[m\times k\right]$ מקבלים מטריצה $AB\left[n\times k\right]$

מה זמן הריצה הנאיבית? לכל משבצת אנו מחשבים סכום של m גורמים, ויש לנו nk משבצות, סה"כ $O\left(nmk\right)$.

נניח שיש 3 מטריצות A,B,C, ורוצים לחשב את ABC. האם זמן הריצה לחישוב $\left(AB\right)C$ שווה לזמן הריצה של $A\left(BC\right)$? לא תמיד. למשל עבור $A\left[7\times3\right],B\left[3\times10\right],C\left[10\times5\right]$, אזי בחישוב $\left(AB\right)C$ יש לנו $7\cdot3\cdot10+7\cdot10\cdot5=210+350=560$, ואילו $A\left(BC\right)$ יש לנו $3\cdot10\cdot5+7\cdot3\cdot5=15\cdot17=255$ פעולות.

קלט: גודלי מטריצות $p_{0},\ldots,p_{n}$ (כאשר $A_{i}\left[p_{i-1}\times p_{i}\right]$).

פלט: המספר המינימלי של פעולות שצריך לבצע ע"מ לחשב את המכפלה $A_{1}A_{2}\cdot\ldots\cdot A_{n}$.

פתרון דינאמי: בשלב האחרון, יש לנו מכפלה כלשהי של $\left(A_{1}\cdot\ldots A_{k}\right)\left(A_{k-1}\ldots A_{n}\right)$. מחיר הפתרון הוא המחיר לחישוב $A_{1}\ldots A_{k}$, בתוספת מחיר חישוב $A_{k+1}\ldots A_{n}$, בתוספת $p_{0}p_{k}p_{n}$.

נגדיר $OPT\left(i,j\right)$ עבור $i\leq j$ להיות מספר הפעולות המינימלי שנדרש כדי לחשב את $A_{i}\cdot\ldots\cdot A_{j}$.

טענה 3.3 קשר רקורסיבי בין הפתרונות האופטימליים: $$OPT_{ij}=\begin{cases}0 & i=j\\ \min_{i\leq k<j}\left\{OPT_{ik}+OPT_{k+1,j}+p_{i-1}p_{k}p_{j}\right\} & i<j\end{cases}$$

הוכחה: עבור $i=j$ ברור. נראה אי שוויון בשני הכיוונים על $i<j$.

1. $OPT_{ij}\leq\min\left\{\ldots\right\}$. יהי k שנותן $\arg\min_{k}\left\{\ldots\right\}$. ניתן לחשב את $A_{i}\ldots A_{j}$ ע"י $\left(A_{i}\ldots A_{k}\right)\left(A_{k+1}\ldots A_{j}\right)$ בעזרת $OPT_{ik}+OPT_{k+1,j}+p_{i-1}p_{k}p_{j}$ פעולות (שמתקבל כי k הוא $\arg\min$).

2. $OPT_{ij}\geq\min\left\{\ldots\right\}$. נתבונן בהשמת סוגריים γ שעלותה OPT_{ij}, ובפרט במכפלה האחרונה שלה, $\left(A_{i}\ldots A_{k^{*}}\right)\left(A_{k^{*}+1}\ldots A_{j}\right)$. עלות γ היא עלות חישוב

$$\begin{aligned}OPT_{ij} & = cost_{\gamma}\left(A_{i}\ldots A_{k^{*}}\right)+cost_{\gamma}\left(A_{k^{*}+1}\ldots A_{j}\right)+p_{i-1}p_{k^{*}}p_{j}\geq\\ & \geq OPT_{ik^{*}}+OPT_{k^{*}+1,j}+p_{i-1}p_{k^{*}}p_{j}\geq\min\left\{OPT_{ik^{*}}+OPT_{k^{*}+1,j}+p_{i-1}p_{k^{*}}p_{j}\right\}\end{aligned}$$

■

האלגוריתם: נמלא טבלה F כאשר f_{ij} יהיה שווה ל־OPT_{ij} בעזרת הנוסחא הרקורסיבית.

$f_{ij}=\min_{i\leq k<j}\left\{f_{ik}+f_{k+1,j}+p_{i-1}p_{k}p_{j}\right\}$ - נמלא את הטבלה בסדר עולה של $j-i$. נחזיר את $f\left(1,n\right)$.

הוכחת נכונות: באינדוקציה על $j-i$ מראים כי $f_{ij}=OPT_{ij}$.

זמן ריצה: יש לנו n^{2} תתי בעיות, ולכן תת־בעיה מחשבים מינימום על $O\left(n\right)$ גורמים, סה"כ $O\left(n^{3}\right)$.

3.3 בעיית מסילות הרכבת

נתונים קטעים שונים של מסילות, ורוצים להרכיב מסילה באורך נתון. לכל מסילה עשוי להיות סוג חיבור שונה בכל אחד מהקצוות, וניתן לחבר רק מסילות בעלי חיבורים משותפים.

קלט: אורך רצוי L. N חלקים $\left\{a_{1},\ldots,a_{N}\right\}$. לכל חלק נתון d_{i} אורך, l_{i} חיבור שמאלי, ו־r_{i} חיבור ימני ששייכים לקבוצת החיבורים $\left\{1,\ldots,p\right\}$, ומחיר c_{i}.

פלט: רשימה של חלקים $t_{1},\ldots,t_{k}$ כך ש־$r_{t_{j}}=l_{t_{j+1}}$ לכל $1<j<k$, סכום $\sum_{j=1}^{k}d_{t_{j}}=L$, ומחיר הפתרון $\sum_{j=1}^{k}c_{t_{j}}$.

מגדירים OPT_k להיות המחיר המינימלי של מסילה באורך k (או ∞ אם לא קיים), עם נוסחת רקורסיה $OPT_k=\min_{1\le i\le N}\left(OPT_{k-d_i}+c_i\right)$ - זה לא יעבוד, צריך עוד תתי־בעיות, כי אולי החיבורים לא מתאימים.

פתרון נכון: נגדיר $OPT_{k,j}$ להיות המחיר המינימלי של מסילה באורך k עם קצה ימני בחיבור j. נוסחת הרקורסיה תהיה $OPT_{kj}=\begin{cases}0 & k=0\\ \min_{1\le i\le a_N, r_i=j, d_i\le k}\left\{OPT_{k-d_i,l_i}+c_i\right\} & o.w.\end{cases}$ (נגדיר $\min\phi=\infty$). זמן הריצה יהיה $O\left(LNp\right)$ לכאורה, אבל לכל k, כשעוברים על כל p המשבצות, מחשבים בסה"כ N סוגים של קטעים. לכן הסיבוכיות היא שוב $O\left(LN\right)$.

הוכחה בע"פ: שלב הבסיס טרוויאלי, להראות את הפתרון האופטימלי - בונים פתרון שזה הערך שלו. לוקחים את הi שמתאים לביטוי, ומראים שניתן לבנות פתרון. בצד השני ניקח פתרון אופטימלי ונראה שהוא לא יותר טוב, כי הוא כלול באיבר במינימום.

האלגוריתם: נמלא טבלה $L\times p$ בסדר עולה של k. בסוף נחזיר את $\min_j OPT_{Lj}$.

הערה 3.4 הפתרון לבעיה הקודמת היה לינארי ב־L. מה היה הקלט? מספר L, רשימה של N חלקים (עם הפרטים שלהם). גודל הקלט הוא $O\left(N+\log L\right)$ ביטים. כלומר הסיבוכיות עשויה להיות אקספוננציאלית בגודל הקלט. בהמשך נדבר על קריפטוגרפיה - כאשר רוצים שמאזינים לא יוכלו לפענח, ממירים את הקוד לפי קוד מוסכם מראש (למשל, לוקחים מספר ראשוני P בן 1000 ספרות). אם אני רוצה לשלוח לו את q, אשלח לו את pq. מי שיאזין, יצטרך לבצע פירוק לגורמים ראשוניים של המספר שקיבל. הסכמה מניחה שהוא לא יוכל, כי אין אלגוריתם יעיל לפרק מספר n לגורמים ראשוניים). למה אין אלגוריתם יעיל לפצוח? הרי אפשר לפרק ב־$O\left(\sqrt{n}\right)$. אבל אם n באורך 1000 ביטים, אז $\sqrt{n}=2^{500}$ - לא יסתיים לעולם. לכן צריך לשים לב שהסיבוכיות היא אקספוננציאלית בגודל הקלט. לפעמים מניחים ש־L נתון בכתיב אונארי כדי לעקוף את הבעיה.

4 אלגוריתם FFT

י"ח כסלו תשע"ב (תרגול 7)

4.1 שחקנים מרכזיים בFFT:

- פולינומים $p\left(x\right)=\sum_{i=0}^{n-1}a_ix^i$ או $\left(a_0,a_1,\ldots,a_{n-1}\right)$.

הגדרה 4.1 עבור $p:=\left(a_0,\ldots,a_{n-1}\right)$ נסמן $p_0:=\left(a_0,a_2,\ldots,a_{n-1}\right),p_1:=\left(a_1,a_3,\ldots,a_{n-2}\right)$.

למה 4.2 $p\left(x\right)=p_0\left(x^2\right)+xp_1\left(x^2\right)$

- שורשי היחידה: פתרונות למשוואה $x^n=1$. מסומנים ω_n^k, כאשר $0\le k\le n-1$,
ו־$\omega_n=\cos\frac{2\pi}{n}+i\sin\frac{2\pi}{n}=e^{\frac{2\pi}{n}i}$. שורשי היחידה מפוזרים על מעגל היחידה במרחקים שווים.

ידע כללי: חזקה של מספר מרוכב: $e^x=1+x+\frac{x^2}{2}+\ldots$, אם נציב $e^{i\theta}$, ניתן להראות שזה מתכנס ל־$\cos\theta+i\sin\theta$.

תכונות:

טענה 4.3 אם n זוגי, אז $\left\{\left(\omega_n^k\right)^2\right\}_{k=0}^{n-1}=\left\{\omega_{\frac{n}{2}}^k\right\}^{\frac{n}{2}-1}$.

הוכחה: $\left(e^{\frac{2\pi k}{n}i}\right)^2 = e^{\frac{2\pi k}{\frac{n}{2}}i} = \begin{cases} \omega_{\frac{n}{2}}^{k} & k < \frac{n}{2} \\ e^{\frac{2\pi}{\frac{n}{2}}\cdot\frac{n}{2}i} e^{\frac{2\pi(k-n/2)}{n/2}} = \omega_{\frac{n}{2}}^{k-\frac{n}{2}} & k \geq \frac{n}{2} \end{cases}$. ■

טענה 4.4 $\sum_{k=0}^{n-1}\left(\omega_n^i\right)^k = \begin{cases} n & i=0 \\ 0 & o.w. \end{cases}$

הוכחה: מהנוסחא של טור גיאומטרי: $\sum_{k=0}^{n-1}\left(\omega_n^i\right)^k = \frac{\left(\omega_n^i\right)^n - 1}{\omega_n^i - 1} = 0$. ■

4.2 התמרת פורייה - אלגוריתם

בהינתן פולינום $p := (a_0, \ldots, a_{n-1})$, רוצים למצוא את $\left(p\left(\omega_n^0\right), p\left(\omega_n^1\right), \ldots, p\left(\omega_n^{n-1}\right)\right)$ עבור $n = 2^k$. ניתן אלגוריתם לרקורסיבי לפתרון הבעיה:

1. אם $n=1$ החזר את a_0.
2. נבנה p_0, p_1.
3. ברקורסיה נחשב את $\left(p_0\left(\omega_{n/2}^0\right), \ldots, p_0\left(\omega_{n/2}^{\frac{n}{2}-1}\right)\right)$, וכנ"ל עבור p_1.
4. עבור $k = 0..n-1$, נחשב את $p\left(\omega_n^k\right) = p_0\left(\left(\omega_n^k\right)^2\right) + \omega_n^k p_1\left(\left(\omega_n^k\right)^2\right) = p_0\left(\omega_{n/2}^k\right) + \omega_n^k p_1\left(\omega_{n/2}^k\right)$.

זמן הריצה הוא $T(n) = O(n) + 2T\left(\frac{n}{2}\right) = O(n\log n)$ בעזרת אינדוקציה או משפט האב. (מימוש נאיבי היה $O\left(n^2\right)$).

דוגמא: $p(x) = 3x^3 + 2x^2 - x - 1$. נרשום את p כ-$(-1,-1,2,3)$.

- בכניסה ראשונה, נחלק: $p_0 := (-1,2), p_1 := (-1,3)$.
- את p_0 נחלק ל-2 $p_{00} = -1, p_{01} = 2$. באופן דומה $p_{10} = -1, p_{11} = 3$.
- נקבל כי $p_{01}(1) = -1, p_{01}(1) = 2, p_{10}(1) = -1, p_{11}(1) = 3$
- נעלה רמה ונחשב

$$\begin{aligned} p_0(1) &= p_{00}(1) + 1p_{01}(1) = -1+2 = 1 \\ p_0(-1) &= p_{00}(1) - 1p_{01}(1) = -1-2 = -3 \\ p_1(1) &= p_{10}(1) + 1p_{11}(1) = -1+3 = 2 \\ p_1(-1) &= p_{10}(1) - 1p_{11}(1) = -1-3 = -4 \end{aligned}$$

• ברמה האחרונה,

$$\begin{aligned} p(1) &= p_0(1) + 1p_1(1) = 1 + 2 = 3 \\ p(i) &= p_0(-1) + ip_1(-1) = -3 - 4i \\ p(-1) &= p_0(1) - 1p_1(1) = 1 - 2 = -1 \\ p(-i) &= p_0(-1) - ip_1(-1) = -3 + 4i \end{aligned}$$

לסיכום: $FFT\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{matrix} FFT\begin{bmatrix} a_0 \\ a_2 \\ \vdots \\ a_{n-2} \end{bmatrix} + \begin{bmatrix} \omega_n^0 \\ \vdots \\ \omega_n^{n/2-1} \end{bmatrix} FFT\begin{bmatrix} a_1 \\ a_3 \\ \vdots \\ a_{n-1} \end{bmatrix} \\ FFT\begin{bmatrix} a_0 \\ a_2 \\ \vdots \\ a_{n-2} \end{bmatrix} + \begin{bmatrix} \omega_n^{n/2} \\ \vdots \\ \omega_n^{n-1} \end{bmatrix} FFT\begin{bmatrix} a_1 \\ a_3 \\ \vdots \\ a_{n-1} \end{bmatrix} \end{matrix}$.

4.3 כפל פולינומים

נתונים $p:(a_0,\ldots,a_{n-1})\,,\ q:(b_0,\ldots,b_{m-1})$. באופן נאיבי יהיה לנו nm פעולות כפל.
תהי s החזקה של 2 הראשונה שגדולה מ־$m+n$. נשלים את p,q לוקטורים באורך s ע"י הוספת אפסים.
נעשה FFT על q,p, נחשב את $pq\left(\omega_s^k\right) = p\left(\omega_s^k\right) q\left(\omega_s^k\right)$ לכל k, ונבצע FFT^{-1} ל־pq.
זמן הריצה יהיה $O(s\log s)$, אבל $s \le 2(m+n)$. נניח בה"כ $n \ge m$, ונקבל
$O(2(m+n)\log 2(m+n)) = O(4n\log 4n) = O(n\log n)$.

4.4 FFT^{-1}

הערה 4.5 $FFT\begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{bmatrix} \omega_n^0 & \omega_n^0 & \ldots & (\omega_n^0)^n \\ (\omega_n^1)^0 & \omega_n^1 & \ldots & (\omega_n^1)^n \\ \vdots & \vdots & \ddots & \vdots \\ (\omega_n^{n-1})^0 & \omega_n^{n-1} & \ldots & (\omega_n^{n-1})^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}$. נסמן $A = VM\left(\omega_n^0, \omega_n^1, \ldots, \omega_n^{n-1}\right)$
נשים לב כי $A(i,j) = \omega_n^{ij}$ עבור $0..n-1$.

הערה 4.6 אם $B = A^{-1}$, אז $\begin{bmatrix} a_0 \\ \vdots \\ a_{n-1} \end{bmatrix} = I\begin{bmatrix} a_0 \\ \vdots \\ a_{n-1} \end{bmatrix} = BA\begin{bmatrix} a_0 \\ \vdots \\ a_{n-1} \end{bmatrix} = B\cdot FFT\begin{bmatrix} a_0 \\ \vdots \\ a_{n-1} \end{bmatrix}$.

טענה 4.7 $A^{-1}(i,j) = \frac{1}{n}\omega_n^{-ij}$.

הוכחה: צ"ל $A^{-1}A = I_n$.

$$\left({}^{"}A^{-1}{}^{"}A\right)(i,j) = \frac{1}{n}\sum_{k=0}^{n-1}\omega_n^{-ik}\cdot\omega_n^{kj} = \frac{1}{n}\sum_{k=0}^{n}\left(\omega_n^{(j-i)}\right)^k = \begin{cases} \frac{1}{n}n = 1 & i = j \\ 0 & i \ne j \end{cases} = I_n(i,j)$$

■

אם נרצה לחשב את FFT^{-1}, נניח וקטור b מתאים לפולינום q (והוא טרנספורם פוריה של p), אזי $A^{-1}b=\frac{1}{n}\begin{bmatrix}q\left(\omega_n^0\right)\\ \cdots\\ q\left(\omega_n^{-(n-1)}\right)\end{bmatrix}$. ניתן במקום זה לעשות FFT על q, נחלק ב־n, ונשנה את הסדר כדי להתאים ל־$A^{-1}b$.

דוגמא: נחשב את FFT^{-1} של $FFT\left(p\right)$.

- בשלב ראשון $p_0:\left(3,-1\right),p_1:\left(-3-4i,-3+4i\right)$
- בשלב הבא $p_{00}:\left(3\right),p_{01}:\left(-1\right),p_{10}\left(-3-4i\right),p_{11}\left(-3+4i\right)$
- כעת נרכיב

$$\begin{aligned}
p_0\left(1\right) &= 3+\left(-1\right)=2\\
p_0\left(-1\right) &= 3-\left(-1\right)=4\\
p_1\left(1\right) &= -6\\
p_1\left(-1\right) &= -8i\\
p\left(1\right) &= 2-6=-4\\
p\left(i\right) &= 4+i\left(-8i\right)=12\\
p\left(-1\right) &= 2+6=8\\
p\left(-i\right) &= 4-i\left(-8i\right)=-4
\end{aligned}$$

נשנה את הסדר ונחלק ב־4, נקבל $\left(-1,-1,2,3\right)$ או הפולינום $p\left(x\right)=3x^3+2x^2-x-1$.

4.5 קונבולוציה

נתונים $a=\left(a_0,\ldots,a_{n-1}\right),b=\left(b_0,\ldots,b_{m-1}\right)$, נסמן $a*b=\left(c_0,\ldots,c_{m+n-2}\right)$, כאשר $c_k=\sum_{i=0}^{k}a_ib_{k-i}$ (אם האינדקס גדול מדי נגדיר $b_{k-i}=0$ וכנ"ל לגבי a)

דוגמא: $a=\left(1,2,3\right),b=\left(-1,1\right)$. אזי $a*b=\left(-1,-1,-1,3\right)$

מסתבר שיש קשר הדוק בין קונבולוציה לכפל פולינומים ־ זה אותו דבר.

למה?

נניח $p:=a,q:=b$, אזי המקדמים של pq יהיו בדיוק $a*b$.

5 אלגוריתמים הסתברותיים

תרגול 8 התבטל עקב שביתה.

6 אלגוריתמי קירוב

ב' טבת תשע"ב (תרגול 9)

בעיית אופטימיזציה:

X הוא אוסף פתרונות חוקיים. $v : X \to \mathbb{R}$ פונקציית ערך לכל פתרון $x \in X$.
רוצים למצוא פתרון $x \in X$ כך ש־$v(x)$ מקסימלי (מינימלי). x כזה קרוי פתרון אופטימלי.

הגדרה 6.1 אלגוריתם יקרא c־מקרב אם הוא מחזיר פתרון חוקי $y \in X$ כך שאם x פתרון אופטימלי, אזי בבעיית מקסימיזציה $cv(y) \geq v(x)$, ובבעיית מינימיזציה $v(y) \leq cv(x)$ (תמיד $c \geq 1$).

6.1 Max Cut

נתון גרף לא מכוון $G = \langle V, E \rangle$. רוצים למצוא חלוקה $V = A \cup B, A \cap B = \phi$, כך ש$|E(A,B)|$ מקסימלי ($E(A,B) = \{\{a,b\} \in E \mid a \in A, b \in B\}$).
אלגוריתם:

- נאתחל $A = V, B = \phi$.
- נמצא קודקוד v כך ש־$|E(v, other\, group)| > |E(v, group(v))|$ - מספר הצלעות מ־v לקבוצה שלו, גדול ממספר הצלעות מ־v לקבוצה השניה.
- נעביר את v לקבוצה השנייה.
- נחזור על התהליך עד שלכל קודקוד v, מספר הצלעות לקבוצה השניה גדול או שווה למספר הצלעות לקבוצה שלו.

טענה 6.2 האלגוריתם הוא 2־מקרב.

הוכחה: מס' הצלעות בפתרון אופטימלי הוא לכל היותר $|E|$. ע"מ להראות שהאלגוריתם 2־מקרב מספיק להראות שמספר הצלעות בחתך $|E(A,B)| \geq \frac{|E|}{2}$.

הערה 6.3 נשים לב כי $\deg v = |E(v, other\, group)| + |E(v, group(v))|$, וכן $2|E| = \sum_{v \in V} \deg v$.

$$\begin{aligned}
2|E| &= \sum_{v \in V} \deg v = \sum_{v \in V} \left(|E(v, group(v))| + |E(v, other\, group)|\right) \leq \sum_{v \in V} 2|E(v, other\, group)| = \\
&= 2 \sum_{v \in V} |E(v, other\, group)| = 2 \cdot 2|E(A,B)| \\
|E(A,B)| &\geq \frac{1}{2}|E|
\end{aligned}$$

■

מספר האיטרציות חסום ע"י $|E|$ כי בכל איטרציה מגדילים את מס' הצלעות בחתך בלפחות 1. בכל איטרציה עוברים על כל השכנים של כל קדקוד - כלומר פעמיים על כל צלע. לכן יש לנו $O\left(|E|^2\right)$.
אלגוריתם הסתברותי הוא טוב אם בהסתברות גבוהה $(1-\varepsilon)$ נותן תשובה גבוהה. אלגוריתם קירוב הסתברותי הוא טוב אם בהסתברות גבוהה $(1-\varepsilon)$ הוא נותן c־קירוב לבעיה.
אלגוריתם:

- לכל $v \in V$, נשים את v ב־A בסיכוי $\frac{1}{2}$, וב־B בסיכוי $\frac{1}{2}$.

- אם רוצים סיכוי לטעות של $\frac{1}{e^k}$, חוזרים על האלגוריתם $k\left(|E|+1\right)$ פעמים, ולוקחים את החתך הגדול ביותר.

אנחנו אומרים שהאלגוריתם טועה אם $|E\left(A,B\right)|<\frac{1}{2}|E|$. נרצה לחסום את הסיכוי לטעות.

1. נחשב את תוחלת מספר הצלעות בחתך: נגדיר מ"מ X להיות מס' הצלעות בחתך. לככל צלע $\{u,v\}\in E$, נגדיר $X_{uv}=\begin{cases}1 & \{u,v\}\in E\left(A,B\right)\\ 0 & o.w.\end{cases}$. אזי $X=\sum_{\{u,v\}\in E}X_{uv}$. $X_{uv}\sim Ber\left(\frac{1}{2}\right)$. לכן $E\left[X\right]=\sum_{\{u,v\}\in E}E\left[X_{uv}\right]=\frac{|E|}{2}$.

2. נשתמש בא"ש מרקוב: יהי X מ"מ אי־שלילי עם תוחלת $E\left[X\right]$, אז $Pr\left(X\geq cE\left[X\right]\right)\leq\frac{1}{c}$. נגדיר מ"מ $Y=|E|-X$. נחסום את $Pr\left(Y\geq\frac{1}{2}|E|\right)$. Y שלם, ולכן $Pr\left(Y\geq\frac{1}{2}|E|\right)=Pr\left(Y\geq\frac{1}{2}|E|+\frac{1}{2}\right)=Pr\left(Y\geq E\left[Y\right]\left(1+\frac{1}{|E|}\right)\right)\leq\frac{|E|}{|E|+1}=1-\frac{1}{|E|+1}$.

הסיכוי לטעות בריצה יחידה $Pr\left(w_1\right)\leq\left(1-\frac{1}{|E|+1}\right)$. אחרי $k\left(|E|+1\right)$ ריצות, הסיכוי שבכל הריצות עשינו טעות הוא $\left(1-\frac{1}{|E|+1}\right)^{k(|E|+1)}=\left(\left(1-\frac{1}{|E|+1}\right)^{|E|+1}\right)^{k}<e^{-k}$.

כל איטרציה לוקחת $O\left(|E|+|V|\right)$. יש $O\left(|E|\right)$ איטרציות (k קבוע). סה"כ $O\left(|E|\left(|V|+|E|\right)\right)$.

ניתן לשלב בין האלגוריתמים ־ במקום להתחיל ב־V,ϕ, אם נתחיל בשתי קבוצות אקראיות, שכבר בחתך שלהם יש מספיק צלעות, נצטרך לעשות רק $O\left(\sqrt{|E|}\right)$ איטרציות. מסתבר שגם כדי לקבל שתי קבוצות אקראיות כאלו מספיק $O\left(\sqrt{|E|}\right)$ חזרות על חלוקה לקבוצות, ולכן בסה"כ יש לנו רק $O\left(|E|\sqrt{|E|}\right)$ בשני השלבים.

6.2 Max 3Sat

SAT ־ נוסחא בוליאנית על אוסף משתנים x_i, היא למשל $\left(x_1\wedge x_2\right)\vee\left(\neg\left(x_3\wedge x_{17}\right)\right)$. השאלה היא האם יש השמה שתוצאתה אמת. למשל $x\wedge\neg x$ לא תסופק לכל השמה.

בהינתן נוסחא, האם יש לה השמה מספקת?

$3SAT$ ־ נאמר שנוסחא היא בצורה $3CNF$ אם היא מכפלת סכומים: $\left(x_1\vee\ldots\vee\neg x_4\right)\wedge\left(\right)\wedge\left(\right)$.

בעיית $3SAT$ זה להחזיר האם לנוסחא ב־$3CNF$ יש השמה מספקת. וגם היא בעיה קשה (נראה כאשר אורך הביטוי בכל סוגריים מוגבל ל3 משתנים).

גירסת אופטימיזציה: $Max\,3\,SAT$

נאמר שפסוקית $\left(x_1\vee x_3\vee x_{15}\right)$ מסופקת אם $x_1=1$ או $x_3=1$ או $x_{15}=1$. נרצה למצוא השמה שמספקת מספר מקסימלי של פסוקיות.

הערה: גם זו בעיה קשה.

הקלט: רשימת משתנים $x_1\ldots x_m$. נוסחא באורך n פסוקיות של 3 משתנים.

אלגוריתם:

לכל משתנה X_i נבחר מתוך $X_i\sim Ber\left(\frac{1}{2}\right)$.

נרצה למצוא השמה שמספקת לפחות $\frac{7}{8}$ מהפסוקיות (בהסתברות $1-\frac{1}{e^k}$).

נגדיר את Y להיות מס' הפסוקיות המסופקות ולכל פסוקית $c_1,\ldots,c_n$. נגדיר Y_i. אזי $E[Y]=\sum E[Y_i]=\frac{7}{8}n$.
נגדיר Z להיות $n-Y$. אזי $Pr\left(Z>\frac{1}{8}n\right)=Pr\left(Z\geq\frac{1}{8}n+\frac{1}{8}\right)=Pr\left(Z\geq\frac{1}{8}n\left(1+\frac{1}{n}\right)\right)\leq 1-\frac{1}{n+1}$.
ולכן נחזור על האלגוריתם $k(n+1)$ פעמים ונחזיר את ההשמה שסיפקה הכי הרבה פסוקיות.
כל איטרציה עולה $O(n+m)$, ויש לנו $O(n)$ איטרציות. סה"כ ז"ר הוא $O(n(n+m))$.

6.3 בעיה קשה

נגדיר את זה בצורה פורמלית בחישוביות. כאן נדבר פחות פורמלית. נסתכל על מרחב הבעיות. P הוא אוסף כל הבעיות שיש להם פתרון פולינומיאלי. NP הוא אוסף הבעיות שניתנות לפתרון ע"י מכונה לא דטרמיניסטית, שמסוגלת לפצל את החישוב שלה. (מחשב 2^n חישובים בזמן ריצה של n). אין כזו מכונה. NP הוא כל הבעיות שהיו יכולות להיפתר בזמן פולינומי ע"י מכונה כזו. בעיקרון המחשבים שלנו יכולים לחשב בזמן אקספוננציאלי בעיות NP. יש בעיות שא"א לפתור - למשל בעיית העצירה (האם תוכנית מחשב תעצור או שיש לה ז"ר אינסופי). ניתן להראות שאם ניתן לפתור בעיות NP "שלמות" (תת-קבוצה של NP) בזמן פולינומיאלי, אז ניתן לפתור את כולן. אין הוכחה אבל ש-$NP \subsetneqq P$. יש גם בעיות שאין לנו פתרון פולינומיאלי, אבל איננו יודעים שהם NP שלמות.

6.4 תת-סכום חלקי.

ט' טבת תשע"ב (תרגול 10 - מאור)

הגדרה 6.4 $FPTAS$ - קירוב $(1\pm\varepsilon)$ לכל $\varepsilon>0$ בזמן פולינומי ב-$n,\frac{1}{\varepsilon}$.

הגדרה 6.5 $PTAS$- קירוב $(1\pm\varepsilon)$ לכל $\varepsilon>0$ קבוע. בזמן פולינומי ב-n.

נניח שזמן הריצה הוא $O\left(n^{2^{\frac{1}{\varepsilon}}}\right)$, זה חוקי ב-$PTAS$, אבל לא ב-$FPTAS$. להיפך, כל מה שהוא $FPTAS$ הוא ב-$PTAS$.
נתונים $a_1,\ldots,a_n\in\mathbb{N}$. $W\in\mathbb{N}$. מחפשים סכום חלקי מקסימלי של $\{a_i\}$ שקטן או שווה ל-W. (מקרה פרטי של knapsack בו $w(x)=\$(x)=x$). כמעט כל $FPTAS$ מגיע מאלגוריתם מדוייק שרץ בזמן ריצה אקספוננציאלי, כאשר אנו מבצעים שיפור שהופך אותו לפולינומיאלי ע"ח הדיוק.
$FPTAS$ לבעיה: יש אלגוריתם נאיבי, לעבור על כל הסכומים החלקיים ולבחור את המקסימלי שקטן או שווה ל-W.
זמן הריצה הוא $O(2^n)$.
רעיון I: נבנה את רשימת הסכומים החלקיים לאט. נזרוק בדרך סכומים שגדולים מ-W.
רעיון II: אם בדרך יש שני תת-סכומים קרובים, נזרוק את הגדול.
למשל: $\{7,51,100,102\}$, $W=200$. נתחיל עם $L_0=\{0\}$ של הקבוצה הריקה. אח"כ $L_1\{0,7\}$ ו-$L_2\{0,7,51,58\}$.
נזרוק את 58 (תלוי מה ה-ε שלי). $L_3=\{0,7,51,100,107,151\}$ נוריד את 107.
$L_4=\{0,7,51,100,107,151,102,109,153,202\}$. נוריד את הסכומים האחרונים, וקבלנו 151.
אלגוריתם:

- נאתחל $L_0=\{0\}$.
- לכל $i=1,\ldots,n$
  - נגדיר $\tilde{L}_i=L_{i-1}\cup(L_{i-1}+a_i)$.

- נגדיר $L_i = \tilde{L_i} \setminus \left(\left\{ l \in \tilde{L_i} \mid l > w \right\} \cup \left\{ l \in \tilde{L_i} \mid \left(1 - \frac{\varepsilon}{n}\right) l \leq l' \leq l, l' \in L_i \right\} \right)$ (ההגדרה לא מדוייקת, אבל המטרה שלכל איבר ב־$\tilde{L_i}$ יישאר איבר ב־L_i שמספיק קרוב.

- נחזיר $\max_{l \in L_n} l$.

איך נקבל את L_i מ$\tilde{L_i}$? נמחק איברים הגדולים מ־W. נעבור על איברי $\tilde{L_i}$ מהקטן לגדול. אם איבר l מקיים שיש איבר גדול ממנו $l' \in \tilde{L_i}$ המקיים $\left(1 - \frac{\varepsilon}{n}\right) l \leq l' \leq l$, נמחק את l.
ברור שהפתרון חוקי. נראה קירוב:

טענה 6.6 לכל i, אם l סכום חלקי של $\{a_k\}_{k=1}^{i}$, אז קיים $l' \in L_i$ כך ש $\left(1 - \frac{\varepsilon}{n}\right)^i l \leq l' \leq l$.

מסקנה 6.7 נפעיל על n, OPT, ונקבל שקיים $l' \in L_n$ כך ש־$\left(1 - \frac{\varepsilon}{n}\right)^n OPT \leq l \leq OPT$. $\left(1 - \frac{\varepsilon}{n}\right)^n \to e^{-\varepsilon}$ מונוטוני, ובפרט $\left(1 - \frac{\varepsilon}{n}\right)^n \geq 1 - \varepsilon$.

הוכחה: באינדוקציה על i.
אם $i = 0$, $L_0 = \{0\}$, והסכום החלקי היחידי הוא 0, והוא נמצא ב־L_0.
נניח נכונות ל־$i-1$. אזי עבור i, יהי $l \leq W$ סכום חלקי של $\{a_1, \ldots, a_i\}$. יש שני מקרים:

1. l סכום חלקי של $\{a_1, \ldots, a_{i-1}\}$. מה"א יש $l' \in L_{i-1}$ כך ש־$\left(1 - \frac{\varepsilon}{n}\right)^{i-1} l \leq l' \leq l$. לכן $l' \in \tilde{L_i}$, ולכן יש $l" \in L_i$ כך ש־$\left(1 - \frac{\varepsilon}{n}\right) l' \leq l" \leq l'$. בסה"כ
$\left(1 - \frac{\varepsilon}{n}\right)^i l = \left(1 - \frac{\varepsilon}{n}\right) \left(1 - \frac{\varepsilon}{n}\right)^{i-1} l \leq \left(1 - \frac{\varepsilon}{n}\right) l' \leq l" \leq l' \leq l$.

2. a_i חלק מהסכום l. ז"א $l - a_i$ סכום חלקי של $a_1, \ldots, a_{i-1}$.
מה"א יש $l' \in L_{i-1}$ כך ש־$\left(1 - \frac{\varepsilon}{n}\right)^{i-1} (l - a_i) \leq l' \leq l - a_i$. לכן $l' + a_i \in \tilde{L_i}$,
ולכן יש $l" \in L_i$ כך ש־$\left(1 - \frac{\varepsilon}{n}\right) (l' + a_i) \leq l" \leq l' + a_i$. בסה"כ $l" \leq l' + a_i \leq l$ וכן

$$\left(1 - \frac{\varepsilon}{n}\right)^i l = \left(1 - \frac{\varepsilon}{n}\right) \left(1 - \frac{\varepsilon}{n}\right)^{i-1} (l - a_i + a_i) \leq \left(1 - \frac{\varepsilon}{n}\right) \left(\left(1 - \frac{\varepsilon}{n}\right)^{i-1} (l - a_i) + a_i \right) \leq \left(1 - \frac{\varepsilon}{n}\right) (l' + a_i) \leq l"$$

(המעבר בזכות זה ש־$1 > \left(1 - \frac{\varepsilon}{n}\right)^{i-1}$).

■

ננתח את זמן הריצה. זמן הריצה לכל איטרציה הוא $O(n |L_n|)$.

טענה 6.8 לכל i, $|L_i|$ פולינומי ב־$n, \frac{1}{\varepsilon}, \log w$.

הוכחה: באינדוקציה על i. נשים לב שלכל היותר יש לנו w מספרים, אבל הם לא צפופים, כי יש לפחות פער של $\left(1 - \frac{\varepsilon}{n}\right)$ ביניהם. נניח שהאיברים ממויינים $l_1 \leq \ldots \leq l_k$. נראה שk פולינומי. $l_1 = 0$ תמיד, לכן $l_2 = \min_j (a_j)$.

$l_3 \geq \frac{l_2}{1-\frac{\varepsilon}{n}}$, $l_4 \geq \left(\frac{l_2}{1-\frac{\varepsilon}{n}}\right)^2$, עד $W \geq l_k \geq \left(\frac{l_2}{1-\frac{\varepsilon}{n}}\right)^{k-2} \geq \frac{1}{\left(1-\frac{\varepsilon}{n}\right)^{k-2}}$. נוציא log ונקבל

$$\begin{aligned}
\log W &\geq -(k-2)\log\left(1-\frac{\varepsilon}{n}\right) \\
k &\leq \frac{\log w}{-\log\left(1-\frac{\varepsilon}{n}\right)}+2 \\
\log\left(1-\frac{\varepsilon}{n}\right) &\leq -\frac{\varepsilon}{n} \\
k &\leq \frac{n\log w}{\varepsilon}+2
\end{aligned}$$

■

6.5 חלוקת מספרים לקבוצות חסומות

נתונים n מספרים בתחום $[0,1]$, המטרה היא לחלקם למינימום קבוצות כך שסכום האיברים בכל קבוצה יהיה קטן מ־1.

רעיון I: כל המספרים שקטנים מ־ε לא יהוו בעיה, נבנה להם קבוצה באופן חמדני.

טענה 6.9 אם כל $\varepsilon \leq a_i$, ויש רק k ערכים שונים, אז אפשר לפתור במדויק.

הוכחה: אם כל $\varepsilon \leq a_i$, אז לכל קבוצה יהיו לכל היותר $\left\lceil\frac{1}{\varepsilon}\right\rceil$ איברים. כמה סוגי קבוצות שונות יש? ניתן להגיע בלי חזרות ל־$X = \binom{\frac{1}{\varepsilon}+k}{k}$. סה"כ מספר החלוקות השונות שיש הוא $\binom{X+n}{X}$, שהוא פולינומי ב־n. ■

נחלק את הקטע $[0,1]$ ל־$k = \left\lceil\frac{1}{\varepsilon^2}\right\rceil$ אינטרוולים. בכל קטע יהיו עד $\lceil n\varepsilon^2\rceil$ איברים שונים. נעביר את כל a_i בכל קטע לאיבר המקסימלי בקטע.

עכשיו אפשר לפתור בזמן פולינומיאת הבעיה החדשה, ולקבל $(1+\varepsilon)$־קירוב.

7 אלגוריתמים קריפטוגרפיים

ט"ז טבת תשע"ב (תרגול 11)

7.1 זמן ריצה של פעולות חשבון.

לא נניח בחלק זה של הקורס שפעולות אריתמטיות מתבצעות ב־$O(1)$.

כן נניח שאפשר להכפיל, לחבר, לחסר, לחלק שני ביטים או ספרות ב־$O(1)$.

דוגמא: ננתח ז"ר של חיבור. נתונים 2 מספרים $m,n \in \mathbb{N}$. נרצה להחזיר את $m+n$.

$n = n_l n_{l-1}\ldots n_0, m = m_k m_{k-1}\ldots m_0$, אזי בזמן ריצה של $O(k+l) = O(\log n + \log m)$.

הערה 7.1 זה פולינומי בגודל הקלט. אלגוריתם שפועל ב־$\Theta(n)$ לא היה פולינומי בגודל הקלט.

כפל: ז"ר $O(\log n \log m)$. בעזרת חיבור הספרות.

מהו זמן הריצה של $\left\lfloor\frac{a}{b}\right\rfloor$ או $a \, mod \, b$? בתרגיל נראה שזה $O(\log a \log b)$.

7.2 מחלק משותף מקסימלי - Greatest Common Denominator

הגדרה 7.2 המספר המקסימלי $gcd\left(a,b\right)$ שמחלק את a ואת b.

הגדרה שקולה: נניח $e_i, f_i \geq 0$, $a=\prod_{i=1}^{n}p_i^{e_i},\, b=\prod_{i=1}^{n}p_i^{f_i}$, אזי $gcd\left(a,b\right)=\prod_{i=1}^{n}p_i^{\min(e_i,f_i)}$.

הגדרה שקולה: $gcd\left(a,b\right)=\min\left(ax+by>0 \mid x,y\in\mathbb{Z}\right)$. למשל עבור $17,23$, יש לנו $3\cdot 23-4\cdot 17=1$.

טענה 7.3 $gcd\left(a,b\right)=\min\left(ax+by>0 \mid x,y\in\mathbb{Z}\right)=\min S$.

הוכחה: נסמן $z=\min S$. $t=gcd\left(a,b\right)$. צ"ל $z=t$.

1. נוכיח $t\leq z$: $t\mid a$ וגם $t\mid b$, לכן $t\mid ax+by$ לכל $x,y\in\mathbb{Z}$. לכן $t\mid z$, ולכן $t\leq z$.

2. $z\leq t$: נראה ש־$z\mid a$ ו־$z\mid b$ ולכן ממקסום t נובע $z\leq t$. $z=ax_0+by_0$. נסתכל על $a\,mod\,z$. נניח בשלילה $0<a\,mod\,z<z$. נראה ש־$a\,mod\,z\in S$, וכך נקבל סתירה למינימליות של z.

$$a\,mod\,z=a-\left\lfloor\frac{a}{z}\right\rfloor z=a-\left\lfloor\frac{a}{z}\right\rfloor\left(ax_0+by_0\right)=a\left(1-\left\lfloor\frac{a}{z}\right\rfloor z_0\right)+b\left\lfloor\frac{a}{z}\right\rfloor y_0$$

נבחר $x_1=1-\left\lfloor\frac{a}{z}\right\rfloor z_0, y_1=\left\lfloor\frac{a}{z}\right\rfloor y_0$ ונקבל $a\,mod\,z=ax_1+by_1$.

■

7.2.1 אלגוריתם אוקלידס

$gcd\left(a,b\right)$ - נניח $a>b$.

- אם $b=0$ תחזיר את a.
- אחרת, תחזיר $gcd\left(b,a\,mod\,b\right)$.

טענה 7.4 $gcd\left(a,b\right)=gcd\left(b,a\,mod\,b\right)$.

הוכחה: $a\,mod\,b=a-\left\lfloor\frac{a}{b}\right\rfloor b$. נראה שאם $z\mid a, z\mid b$ אז $z\mid a\,mod\,b$: הוא מחלק את שני המרכיבים של $a\,mod\,b$. נניח $z\mid b$ וגם $z\mid a\,mod\,b$ ונראה $z\mid a$: $a=a\,mod\,b+\left\lfloor\frac{a}{b}\right\rfloor b$, ולכן $z\mid a$. הראינו שקבוצת המחלקים המשותפים של a,b שווה לקבוצת המחלקים המשותפים של $a\,mod\,b, b$, ולכן המקסימום של הקבוצות שווה. ■

זמן הריצה של האלגוריתם. חישוב $a\,mod\,b$ לוקח $O\left(\log a\log b\right)$. השאלה היא כמה איטרציות gcd יהיו.

טענה 7.5 יהיו $O\left(\log b\right)$ קריאות רקורסיביות.

הוכחה: תזכורת: בסדרת פיבונצ'י $\frac{\Phi^k}{\sqrt{5}}-1\leq F_k\leq\frac{\Phi^k}{\sqrt{5}}+1$.

למה 7.6 נניח שהיו kקריאות רקורסיביות, אזי $a\geq F_{k+1}, b\geq F_k$. באינדוקציה על k:

בסיס: $k=1$, $b\geq 1=F_1$, $a\geq 2=F_2$.

מעבר: אם היא k קריאות ל־a,b, היו $k-1$ קריאות ל־$b,a\,mod\,b$. מה"א $b\geq F_k, a\,mod\,b\geq F_{k-1}$. $a\,mod\,b=a-\left\lfloor\frac{a}{b}\right\rfloor b$, ולכן $a=a\,mod\,b+\left\lfloor\frac{a}{b}\right\rfloor b\geq F_{k-1}+\left\lfloor\frac{a}{b}\right\rfloor F_k\geq F_{k-1}+F_k=F_{k+1}$

מסקנה 7.7 בהינתן b. נבחר k שמקיים $F_{k-1} \leq b < F_k$. אם קוראים לאלגוריתם עבור a, b, מספר הקריאות הרקורסיביות קטן מ־k. אחרת $b \geq F_k$. לכן אנו צריכים להראות כי $k = O\left(\log F_k\right)$ מהנוסחה בתזכורת, ונקבל שמספר האיטרציות הוא $O\left(\log b\right)$.

■

מסקנה 7.8 זמן הריצה של אלג' אוקלידס הוא $O\left(\log b\right) O\left(\log a \log b\right) = O\left(\log^2 b \log a\right)$.

7.2.2 אלגוריתם אוקלידס המורחב

בהינתן a, b, יש $x, y \in \mathbb{Z}$ כך ש־$gcd\left(a, b\right) = ax + by$. אלגוריתם אוק' המורחב מוצא את x, y.
נניח שיש לי x_0, y_0 כך ש־$x_0 b + y_0 \left(a \, mod \, b\right) = gcd\left(b, a \, mod \, b\right) = gcd\left(a, b\right)$. נחפש x, y כך ש־$xa + yb = gcd\left(a, b\right)$.

$$gcd\left(a, b\right) = x_0 b + y_0\left(a - \left\lfloor \frac{a}{b} \right\rfloor b\right) = y_0 a + \left(x_0 - \left\lfloor \frac{a}{b} \right\rfloor y_0\right) b$$

ולכן ניתן לקחת $x = y_0, y = x_0 - \left\lfloor \frac{a}{b} \right\rfloor y_0$.
האלגוריתם:

- אם $b = 0$, החזר $x = 1, y = 0, gcd\left(a, b\right) = a$.
- חשב את $gcd - ext\left(b, a \, mod \, b\right)$ וקבל $x_0, y_0, gcd\left(b, a \, mod \, b\right)$. החזר את $x = y_0, y = x_0 - \left\lfloor \frac{a}{b} \right\rfloor y_0, gcd\left(a, b\right) = gcd\left(b, a \, mod \, b\right)$.

דוגמא:

$$\begin{array}{rl}
(42, 25) & x = 3, y = -2 - 1 \cdot 3 = -5 \\
(25, 17) & x = -2, y = 1 - 1\left(-2\right) = 3 \\
(17, 8) & x = 1, y = 0 - 2 \cdot 1 = -2 \\
(8, 1) & x = 0, y = 1, gcd = 1 \\
(1, 0) & x = 1, y = 0, gcd = 1
\end{array}$$

7.3 משפט השאריות הסיני

יהיו $n_1, \ldots, n_k$ k מספרים זרים זה לזה. נגדיר $n = \prod_{i=1}^{k} n_i$. נתבונן בפונקציה $f : \mathbb{Z}_n \to \mathbb{Z}_{n_1} \times \ldots \times \mathbb{Z}_{n_k}$ המוגדרת ע"י $f\left(x\right) = \left(x \, mod \, n_i\right)_{i=1}^{k}$.

משפט 7.9 f חח"ע ועל.

הוכחה: נשים לב כי $\left|\mathbb{Z}_n\right| = \left|\mathbb{Z}_{n_1} \times \ldots \times \mathbb{Z}_{n_k}\right|$, ולכן מספיק להראות ש־$f$ על. נראה שלכל וקטור שאריות ב־$\mathbb{Z}_{n_1} \times \ldots \times \mathbb{Z}_{n_k}$ יש אלגוריתם למציאת איבר ב־$\mathbb{Z}_n$ שאלו השאריות שלו. ■

שלב 1: נמצא מספרים $z_i \in \mathbb{Z}_n$ כך ש־ $z_i \, mod \, n_j = \begin{cases} 1 & i = j \\ 0 & o.w. \end{cases}$.

נתחיל מלהגדיר $m_i = \frac{n}{n_i} = \prod_{j\neq i} n_j$. אזי $m_i \, mod \, n_j = 0$ עבור $j \neq i$, אבל לא בטוח ש־$m_i \, mod \, n_i = 1$. נסתכל על $m_i \, mod \, n_i \in \mathbb{Z}_n^*$ (כי n_j זר ל־n_i לכל $j \neq i$, ולכן המכפלה m_i זרה גם היא). לכן יש $r_i = (m_i \, mod \, n_i)^{-1}$. נגדיר $z_i = r_i m_i \, mod \, n$.

איך מוצאים הופכי ב־$\mathbb{Z}_n^*$? בהינתן $a \in \mathbb{Z}_n^*$, רוצים $b \in \mathbb{Z}_n^*$ המקיים $ab \, mod \, n = 1$, כלומר $ab + kn = 1$, אם נחפש b, k המקיימים זאת, b יהיה ההופכי של a ב־$\mathbb{Z}_n^*$.

שלב 2: $a = \sum_{i=1}^{k} z_i a_i \, mod \, n$. נסתכל על

$$a \, mod \, n_j = \left(\sum_{i=1}^{k} z_i a_i \, mod \, n\right) mod \, n_j = \left(\sum_{i=1}^{k} (z_i a_i \, mod \, n_j)\right) mod \, n_j = z_j a_j \, mod \, n_j = a_j$$

7.4 תזכורת ־ חבורות

כ"ג טבת תשע"ב (תרגול 12)

חבורה ־ $(G, \cdot)$ המקיימת אסוציאטיביות, איבר יחידה, הופכי.

דוגמא: $(\mathbb{Z}_n, + \, mod \, n)$. או $(\mathbb{Z}_n^*, \cdot \, mod \, n)$. מהו $|\mathbb{Z}_n^*|$? אם $n = \prod_{i=1}^{n} p_i^{e_i}$ אז $\varphi_n = n \prod_{i=1}^{p} \left(1 - \frac{1}{p_i}\right) = \prod_{i=1}^{n} p_i^{e_i} (p_i - 1)$.

אם יש לנו תת־חבורה $H \leq G$, אז $|H| \mid |G|$.

מסקנה 7.10 לכל $g \in G$, הסדר של g, $ord\,(g) = \min\left\{k \mid g^k = e\right\}$ מקיים $ord\,(g) \mid |G|$.

דוגמא: עבור $\mathbb{Z}_7 = \{1,2,3,4,5,6\}$, נקבל $ord\,(2) = 3 \mid 6$.

למה לאיבר יש בהכרח סדר? יהי $g \in G$, נסתכל על $\{g^i \mid i \in \mathbb{N}\} \subseteq G$. אבל G סופית, לכן קיימים $g^k = g^l$. נניח $l < k$. נכפיל את המשוואה ב־$(g^{-1})^l$, ונקבל $g^{k-l} = g^k g^{-l} = g^l g^{-l} = e$.

הוכחה: נתבונן ב־$\left\{e, g, g^2, \ldots, g^{ord(g)-1}\right\}$. זו תת־חבורה של G, ולכן גודלה מחלק את $|G|$ (משפט לגרנז'), כלומר $ord\,(g) \mid |G|$. ∎

הגדרה 7.11 חבורה ציקלית היא חבורה סופית G עם איבר $g \in G$ מסדר $|G|$. g כזה ייקרא יוצר של החבורה.

בדוגמא הקודמת, $ord\,(3) = 6$ כי $3^3 = 27_7 = 6 \neq 1$. $ord\,(4) = 3$, $ord\,(5) = 6$ ו־$ord\,(6) = 2$.

משפט 7.12 (ללא הוכחה) אם p ראשוני, $d \mid (p-1)$, מספר האיברים ב־$\mathbb{Z}_p^*$ מסדר d זה בדיוק $\varphi\,(d)$. בפרט $\mathbb{Z}_p^*$ ציקלית לכל $p \in P$ ראשוני.

דוגמא ל־$\mathbb{Z}_n^*$ שאינו ציקלי: $\mathbb{Z}_8^* = \{1,3,5,7\}$, נקבל $ord\,(g) = 2$ לכל $g \neq 1$.

7.5 הצפנה ־ דוגמאות

המטרה: להעביר מידע $A \to E \to B$, בלי ש־E יוכל לדעת מה המידע.

1. נניח A, B מסכימים מראש על מחרוזת $x \in \{0,1\}^k$ סודי. נניח שA רוצה לשלוח הודעה לB מסוג $m \in \{0,1\}^k$. אם $x = x_{k-1}x_{k-2}\ldots x_0$, ו־$m = m_{k-1}\ldots m_0$, הוא ישלח את $(x_{k-1} \oplus m_{k-1}, \ldots, x_0 \oplus m_0)$ (bitwise XOR). B יקבל את ההודעה, ויפעיל עליה שוב $Bitwise\,XOR$, ויקבל את m.

2. נניח ש־A, B מסכימים על מספר ראשוני גדול p. נניח שA רוצה לשלוח ל־B מספר ראשוני גדול q. A שולחת את pq, B יוכל לפענח אם הוא יחלק ב־p. E יוכל לעשות זאת אם יפרק את pq לגורמים, אבל אם אין לו דרך יעילה לעשות זאת, הוא לא יצליח (הנחה: לפרק מספר לגורמים ראשוניים זה קשה).

7.6 אלגוריתם דיפי־הלמן

A, B ־ רוצים להסכים על מספר סודי x, כך שמי שמאזין לא יודע את x.

- A מגרילה מספר ראשוני גדול p, ומספר $g \in \mathbb{Z}_p^*$, כך ש־g יוצר של $\mathbb{Z}_p^*$ (בהמשך נראה איך), ושולחת את (p, g) ל־B. (כל העולם יודע את (p, g)).
- A מגרילה מספר סודי $a \in \mathbb{Z}_p^*$ ושולחת ל־B את $g^a \in \mathbb{Z}_p^*$.
- B מגריל מספר $b \in \mathbb{Z}_p^*$, ושולח ל־A את g^b.
- A מחשבת את $\left(g^b\right)^a = g^{ab}$. באופן דומה B יחשב את $(g^a)^b = g^{ab}$, וזה יהיה המספר הסודי.

מי שמאזין יודע את $\left(p, g, g^a, g^b\right)$ וצריך למצוא את g^{ab}. בעיה זו נקראת בעיית דיפי־הלמן, ולא ידוע אלגוריתם יעיל שפותר אותה.
בעיה קשורה: בעיית הלוג הדיסקרטי. בהינתן p, g, g^a, למצוא את a (יותר קשה מDH).
מציאת ראשוני גדול: נלמד בהמשך אלגוריתם מילר־רבין, שבהינתן מספר בודק אם הוא ראשוני. הצפיפות של מספרים ראשוניים בגודל n (בסביבות $\log n$ ספרות) הוא בערך $\frac{1}{\ln n}$.
איך מוצאים יוצר? נרצה לדעת את הגורמים הראשונים של $(p-1)$, למשל אפשר להגריל ראשוני q גדול ולהסתכל על מספרים מצורה $2kq+1$
דוגמא: אם $q = 13$, נקבל ש־27 אינו ראשוני, אבל 53 ראשוני. ו־$52 = 2^2 \cdot 13$.

טענה 7.13 g יוצר אס"ם לכל $p-1 \mid q \neq 1$, $g^{\frac{p-1}{q}} \neq 1$.

הוכחה: מימין לשמאל טרוויאלי. משמאל לימין: $ord(g) \mid p-1$. אם $ord(g) < p-1$, אז קיים q כך ש־ $ord(g) = \frac{p-1}{q}$. ■

נבדוק אם g יוצר ע"י חישוב $g^{\frac{p-1}{z}}$ לכל ראשוני z שמחלק את $p-1$ (זה מספיק).
נמצא יוצר ע"י הגרלת איברים ב־$\mathbb{Z}_p^*$ ובדיקה האם הם יוצרים. (לא נוכיח, אבל $\varphi(p-1) = \Omega\left(\frac{p}{\log\log p}\right)$, כלומר לא נצטרך לנסות הרבה).

7.7 חלוקת סוד

נניח שיש קוד סודי לפצצה, ורוצים לחלק אותו בין n אנשים כך שכולם ביחד יוכלו להפעיל את הפצצה, אבל כל צירוף חלקי שלהם לא יוכל.
נניח שהקוד הוא $f_0 \in \mathbb{Z}_p$ עבור p ראשוני גדול.
נגריל מספרים $f_1, \ldots, f_{n-1} \in \mathbb{Z}_p$ שונים זמ"ז ומאפס. נגדיר פולינום $f(x) = \sum_{i=0}^{n-1} f_i x^i$ מעל $\mathbb{Z}_p$.
נגריל עוד n מספרים $u_0, \ldots, u_{n-1}$ שונים, וניתן למדינה k את הקוד $(u_k, f(u_k))$.

אם n המדינות משתפות פעולה, לכל מדינה יש משוואה ליניארית (והחזקות שלו ידועות u_k) עם n משתנים (המקדמים של f). בעזרת כל המדינות ניתן לפתור את מקדמי הפולינום, ולגלות את ערך f_0 הדרוש ע"י פתרון n משוואות ב־n נעלמים.
אם $n-1$ מדינות משתפות פעולה, יש להן $n-1$ משוואות, ו־n נעלמים, והן לא יכולות לפתור (יש p פתרונות אפשריים).
בתרגיל צריך להכליל ל־n מדינות עם k מדינות שמשתפות פעולה.

7.8 הצפנת RSA

ר"ח שבט תשע"ב (תרגול 13)
יש k משתתפים. לכל משתתף יש מפתח פומבי ומפתח סודי.
המפתח הפומבי משמש לקידוד הודעות, והמפתח הסודי לפיענוח הודעות.
נסמן: M_i ־ מרחב ההודעות של המשתתף ה־i.
לכל משתתף יש פונקציית קידוד $E_i : M_i \to M_i$ ופונקציית פענוח $D_i : M_i \to M_i$. E_i פומבי, D_i סודי, ו־$D_i = E_i^{-1}$. ($D_i \circ E_i = E_i \circ D_i = id$).
נניח A רוצה לשלוח הודעה m ל־B, היא תשלח את $E_B(m)$, ו־B יוכל לפענח ע"י $D_B(E_B(m)) = m$.
המאזין יודע את $E_B(m) \in M_B$, ואת E_B, הסכמה תהיה בטוחה אם E_B קשה להפיכה.
RSA־ נניח שברוך רוצה להצטרף למשתפים. הוא צריך להגדיר D, E, M.

1. נגריל 2 ראשונים גדולים p, q.
2. נחשב את $n = pq, \varphi(n) = (p-1)(q-1)$.
3. נמצא מספר $e \in \mathbb{Z}^*_{\varphi(n)}$.
4. נחשב את $d = e^{-1} \in \mathbb{Z}^*_{\varphi(n)}$ ($ed = 1\,(mod\,\varphi(n))$).

המפתח הפומבי יהיה n, e. המפתח הסודי זה d.
מרחב ההודעות הוא $\mathbb{Z}^*_n$. פונקציית הקידוד עבור $m \in \mathbb{Z}^*_n$ היא $E(m) = m^e\,mod\,n$, ופונקציית פענוח $D(m) = m^d$.
נקבל כי $D(E(m)) = D(m^e) = (m^e)^d = m^{ed} = m^{k\varphi(n)+1} = \left(m^{\varphi(n)}\right)^k m = 1 \cdot m = m$, ובאופן דומה $E(D(m)) = m$.
השאלה היא למה קשה לשבור את הקוד.

- אם יודעים לפקטר ־ שוברים את הקוד.
- אם יודעים להוציא שורש מסדר e ב־$\mathbb{Z}^*_n$, ניתן לשבור את הקוד.

דוגמאות לRSA לא בטוח:

1. נניח ש־e קטן, ו־m קטן כך ש־$m^e < n$. ניתן להוציא שורש מעל $\mathbb{Z}$.
2. נניח שצור, שולמית ומאור מצטרפים לסכמה. כל אחד בוחר ראשוניים שונים $p_1, q_1, p_2, q_2, p_3, q_3$, אבל כולם בוחרים $e = 3$. יובל שולח לכולם את המבחן מקודד. הסטודנט המאזין יודע את $m^3\,mod\,n_1, m^3\,mod\,n_2,\ m^3\,mod\,n_3$. לפי משפט השאריות הסיני, ניתן לחשב את $m^3\,mod\,n_1n_2n_3$, אבל $m^3 < n_1n_2n_3$, ועכשיו ניתן להוציא שורש.

דוגמא להרצת RSA:

נבחר $p=11,q=7$ ולכן $n=77,\varphi\left(n\right)=60$.

נבחר $e=7$. נחשב את e^{-1}: $gcd\left(60,7\right)=gcd\left(7,4\right)=gcd\left(4,3\right)=gcd\left(3,1\right)=gcd\left(1,0\right)$, ולכן $1=1\cdot1+0\cdot0$, בשלב הבא $1=0\cdot3+1\cdot1$, $1=1\cdot4-1\cdot3$, $1=-1\cdot7+2\cdot4$, $1=2\cdot60-17\cdot7$. כלומר $d=e^{-1}=43$ בשדה $\mathbb{Z}_{60}^{*}$.

נקודד את $m=10$: $10^{7}=10000000\,mod\,77=10$. נחשב את $10^{43}\,mod\,77=\left(10^{7}\right)^{6}\cdot10=10^{6}\cdot10$

ניקח $p=11,q=23$, אזי $n=253,\varphi\left(253\right)=220$.

ניקח $e=3$. נחשב את $d=e^{-1}=147$.

נבחר הודעה $m=43$. נקודד $m^{3}=43^{3}=65$. נפענח ע"י $65^{2}=177$, וכן הלאה.

7.8.1 חתימה דיגיטלית

בוב רוצה לזהות שההודעה אכן הגיעה מאליס. אליס ממציאה חתימה σ, ושולחת לבוב את $D_{A}\left(\sigma\right)$ (פונקציית הפענוח של אליס על σ). בוב יכול לחשב את $E_{A}\left(D_{A}\left(\sigma\right)\right)=\sigma$, וכך הוא יודע שאליס שלחה את ההודעה.

7.9 אלגוריתם רבין־מילר

בהינתן מספר אי־זוגי n נרצה לדעת האם הוא ראשוני.

נעביר את n בשני מבחנים הסתברותיים, אם הוא נופל באחד משני המספרים האלו נאמר שהוא פריק, ואם הוא עובר, נאמר שהוא ראשוני. צריך להראות שההסתברות להצהרה מוטעית כזו קטנה מחצי.

אם n ראשוני, האלגוריתם תמיד יחזיר שהוא ראשוני, אם n פריק, האלגוריתם יחזיר פריק בהסתברות של לפחות חצי.

המבחנים מתבססים על שתי תכונות של מספרים ראשוניים:

1. משפט פרמה הקטן: $a^{p-1}=1\left(mod\,p\right)$
2. אם p ראשוני, $\mathbb{Z}_{p}$ שדה, ולמשוואה $x^{2}=1$ יש בדיוק שתי פתרונות: $1,p-1$.

האלגוריתם יבחר a, וינסה להפיל את המספר באחד משני המבחנים.

1. אם n זוגי נחזיר פריק.
2. נגריל $1<a<n$ בהסתברות אחידה. אם $gcd\left(a,n\right)\neq1$, נחזיר פריק.
3. נחשב את $a^{n-1}\,mod\,n$. אם זה לא 1 נחזיר פריק.
4. נרשום $n-1=2^{s}u$ כאשר u אי־זוגי. אנו מניחים כי $a^{n-1}=1$.
5. נחשב את $a^{u},a^{2u},a^{4u},\ldots,a^{2^{s}u}=1$. יהי j המינימלי כך ש־$a^{2^{j}u}=1$. אם $j=0$ נחזיר ראשוני.
6. נבדוק האם $a^{2^{j-1}u}\in\left\{ -1,1\right\}$. אם כן נחזיר ראשוני, אחרת נחזיר פריק.

8 חזרה

ח' שבט תשע"ב (תרגול 14)

8.1 קונבולוציה - שימוש

נתון פולינום $p(x)=\sum_{i=0}^{n}p_i x^i$ מעל $\mathbb{R}$. נתון $a\in\mathbb{R}$. ונרצה למצוא את מקדמי הפולינום $p(x+a)$ בזמן $O(n\log n)$.

דוגמא: $p(x)=3x^2+2x+1$, אזי $p(x+4)=3(x+4)^2+2(x+4)+1=3x^2+26x+57$, ונחזיר $(3,26,57)$.

אלגוריתם נאיבי: מציב $p(x+a)=\sum_{i=0}^{n}p_i(x+a)^i$, פותח את הסוגריים ומכנס איברים. כל גורם $(x+a)^i$ נפתח ל־$i+1$ איברים לפי נוסחת הבינום של ניוטון, ולכן סה"כ נקבל $\sum_{i=0}^{n}(i+1)=\Theta(n^2)$ איברים שצריך לכנס אותם.

אלגוריתם שמשתמש בקונבולוציה: נגדיר $b=\{b_i\}_{i=0}^{n}$ ע"י $b_i=\frac{x^i}{i!}$. $c=(c_j)_{j=0}^{n}$ ע"י $c_i=p_i i!$, ו־$d=(d_i)_{i=0}^{n}$ ע"י $d_i=\frac{a_i}{i!}$. נסמן $c_i^{rev}=c_{n-i}$.

$$\begin{aligned}
p(x+a) &= \sum_{i=0}^{n}p_i(x+a)^i=\sum_{i=0}^{n}p_i\sum_{j=0}^{i}\binom{i}{j}x^j a^{i-j}=\sum_{i=0}^{n}\sum_{j=0}^{i}p_i\binom{i}{j}x^j a^{i-j}=\\
&= \sum_{j=0}^{n}\sum_{i=j}^{n}p_i\binom{i}{j}x^j a^{i-j}=\sum_{j=0}^{n}\sum_{i=j}^{n}p_i\frac{i!}{j!\,(i-j)!}x^j a^{i-j}=\sum_{j=0}^{n}\frac{x^j}{j!}\sum_{i=j}^{n}p_i\frac{i!}{(i-j)!}a^{i-j}\\
&= \sum_{j=0}^{n}b_j\sum_{i=j}^{n}c_i d_{i-j}=\sum_{j=0}^{n}b_j\sum_{i=0}^{n-j}c_{i+j}d_i=\sum_{j=0}^{n}b_j\sum_{i=0}^{n-j}c_{n-i-j}^{rev}d_i=\sum_{j=0}^{n}\frac{x^j}{j!}\left(c^{rev}*d\right)_{n-j}
\end{aligned}$$

לכן ניתן להשתמש באלגוריתם הבא:

1. נחשב את הסדרות c^{rev},d ב־$O(n)$.
2. נחשב את הקונבולוציה $c^{rev}*d$ ב־$O(\log n)$.
3. נחזיר את המקדם ה־j להיות $\frac{(c^{rev}*d)_{n-j}}{j!}$.

8.2 אלגוריתמי קירוב

מועד א' תשס"ח:

קלט: גרף מכוון $G=(V,E)$.

פלט: תת־קבוצה של E שלא מכילה מעגל, וגודלה מקסימלי.

דרוש אלגוריתם 2־מקרב.

רמז: מספר את הקודקודים בסדר כלשהו.

אלגוריתם: נמספר את הקודקודים בסדר כלשהו $v_1,\ldots,v_n$.

נגדיר 2 קבוצות סדרות $E_1=\{(v_i,v_j)\mid i<j\},E_2=\{(v_i,v_j)\mid i>j\}$ כך ש־$E_1\coprod E_2=E$. נחזיר את המקסימלית ביניהם.

הוכחת נכונות:

<u>חוקיות</u>: עבור E_1, נניח בשלילה שיש מעגל $v_{i_1},\ldots,v_{i_k}$. מהגדרת E_1 נובע $v_{i_1}<v_{i_2}<\ldots<v_{i_k}<v_{i_1}$ שזו סתירה. באופן דומה עבור E_2.

<u>אופטימליות</u>: מתקיים $sol=\max\{|E_1|,|E_2|\}\geq\frac{1}{2}|E|\geq\frac{1}{2}OPT$.

המשך השאלה: לכל $n>1$ אי זוגי, תן דוגמא לגרף על n קודקודים עבורו קיים מספר שעבורו האלגוריתם מחזיר פתרון אופטימלי, וגם קיים מספר עבורו האלגוריתם מחזיר פתרון גרוע פי שניים.

8.3 זרימה ברשתות

קלט: n אנטנות סלולריות שנתונות ע"י מיקומן במישור. $b_1, \dots, b_n$.
n מכשירים סלולריים הנתונים ע"י מיקומם ברגע נתון. $p_1, \dots, p_n$.
נתון r, רדיוס הפעולה של האנטנות.

הגדרה 8.1 נאמר שהרשת ניתנת לחיבור במלואה אם לכל טלפון יש אנטנה במרחק קטן מ־r ממנו, וגם לכל אנטנה מחוברים לכל היותר 3 טלפונים.

1. כתוב אלגוריתם המכריע ב־$O\left(n^3\right)$ האם הרשת ניתנת לחיבור במלואה.

בגדול הרעיון להשתמש ברשת דו צדדית הבאה: הפלאפונים מחוברים למקור, האנטנות מחוברות למטרה, הפלאפונים מחוברים לאנטנות המתאימות. הצלעות באמצע ומהמקור הן בקיבולת 1, ומהאנטנות למטרה בקיבול 3.

לשים לב לנקודות הבאות:

(א) בבעיות הכרעה, צריך להוכיח בהוכחת נכונות שני כיוונים: אם קיים שיבוץ חוקי, נמצא זרימה בשטף הדרוש. אם קיימת זרימה בשטף n, קיים שיבוץ חוקי.

(ב) בניתוח זמן ריצה, להשתמש במונחים המקוריים ולא במונחים של הגרף. במונחים של n.

למה הסיבוכיות היא $O\left(n^3\right)$? כי מספר האיטרציות קטן יותר.

2. נניח שנתונה רשת סלולרית הניתנת לחיבור במלואה. נניח שמתבצעת סדרה של n עדכונים, כאשר בכל עדכון אחד הטלפונים זז למיקום חדש, והשאר נשארים במקומם. תן אלגוריתם שמכריע עבור כל עדכון האם הרשת עדיין ניתנת לחיבור במלואה, ואם כן, מחזיר את השיבוץ (אם לא ־ יוצא, ולא ממשיך עדכונים נוספים).

פתרון:

(א) נריץ את האלגוריתם מסעיף א' ב־$O\left(n^3\right)$ ונמצא שיבוץ (נתון שקיים שיבוץ כזה). נשמור את רשת הזרימה.

(ב) עבור על עדכון של p_i נבצע:

i. נוריד את הזרימה לאורך המסלול מ־s ל־t העובר ב־p_i וזורם בו 1, מ־1 ל־0.

ii. נעדכן את החצים שיוצאים מ־p_i.

נשים לב: יש לנו ביד רשת זרימה שמתאימה למצב הנוכחי לאחר העדכון של p_i, וזורמת בה זרימה בשטף $n-1$.

iii. נריץ איטרציה אחת של $Edmond - Karp$.

iv. אם הזרימה גדלה ב־1, קבלנו זריה משטף n, ונחזיר שאפשר, ואת השיבוץ, ונחזור לב'.

v. אם הזרימה לא גדלה, נחזיר שאי אפשר ונצא.

הוכחת נכונות: עלינו להראות שלאחר איטרציה אחת נקבל את הזרימה המקסימלית ברשת הזרימה שמתאימה למצב הנוכחי (השאר הוכחנו בסעיף א').

יש זרימה בגודל $n-1$. הזרימה המקסימלית יכולה להיות בשטף $n-1$ או n. ברור שאם הזרימה המקסימלית היא משטף n, האיטרציה הנוכחית תמצא מסלול משפר ברשת השיורית, ותעלה את הזרימה לפחות ב־1, ונקבל זרימה משטף n. אם הזרימה היא משטף $n-1$, היא לא תשפר, ונשאר עם זרימה מקסימלית.

זמן ריצה: איטרציה של EK עולה BFS שזה $O\left(|V|+|E|\right)$, וחוזרים עליה n, בסה"כ $|E|=O\left(n^2\right)$, ולכן יש לנו $O\left(n^3\right)$.

8.4 אלגוריתמים דינאמיים

דוגמא לשימוש בעייתי בתכנון דינמי:

קלט: גרף לא מכוון דו"צ. $G=\left(L\cup R,E\right)$. פונקציית משקל $w:E\to\mathbb{R}_+$.

פלט: משקל של התאמה בעלת משקל מקסימלי בגרף.

פתרון (בעייתי) באמצעות תכנון דינאמי:

לכל $U\subseteq E$ נגדיר $OPT\left(U\right)$ להיות המשקל המקסימלי של התאמה שמשתמשת רק בצלעות מתוך U.

נרצה למצוא את $OPT\left(E\right)$.

נשים לב שמתקיים הקשר הרקורסיבי: $OPT\left(U\right)=\begin{cases}0 & U=\phi\\ \max_{e\in U}\left\{w\left(e\right)+OPT\left\{e'\in U\mid e'\cap e=\phi\right\}\right\} & U\neq\phi\end{cases}$.

ניתן להוכיח את נוסחת הרקורסיה.

החיסרון: האלגוריתם דורש מעבר על כל תתי־הבעיות.

8.5 המבחן

בעבר הופיעו 4/5 או 3/4 שאלות. שנה שעברה המבנה שונה משאלות מרובות סעיפים לפיצול של הסעיפים הקטנים לחלק א', וחלק נוסף של שאלות מחשבה. השנה עוד לא החליטו.