אלגוריתמים 2

18 בפברואר 2016

מרצה: ד"ר עדן כלמטץ'
מסכם: מני סדיגורסקי

סיבוכיות של פעולות אריתמטיות

29.10.15

תזכורת: חיבור וחיסור ניתנים לביצוע בזמן $\mathcal{O}(n)$ כאשר n הוא מספר הביטים (ולא גודל המספר). כפל וחילוק ניתנים לחישוב בזמן $\mathcal{O}(n^2)$.
כלומר אם אנו עובדים עם $m_1, m_2 \approx 1024$ אזי חיבור, לדוגמה, יתבצע בייצוג בינארי בזמן $\mathcal{O}(10) = \mathcal{O}(log_2(1024))$ ולא בזמן $\mathcal{O}(1024)$.

הערה: חשוב לשים לב שמצד שני אם נבנה אלגוריתמים שרצים בזמן ריצה שתלוי בגודל הקלט (כלומר גודל המספר, נניח 1024) אזי התלות באורך הקלט כלומר (גודל הייצוג של הקלט נניח, 10 ביטים) תהיה גדולה אקספוננציאלית.
לדוגמה בבעית הגנב $(Knapsack\ problem)$ הראנו שניתן בעזרת תכנון דינאמי לבנות אלגוריתם שרץ בזמן שהוא לינארי בגודל המספרי של הקלט m והיה נראה לנו שזה מצוין אלא שבעצם מה שחשוב בדרך כלל זה הייצוג כי זה אורך הקלט של התוכנית ובמקרה הזה נקבל שזמן הריצה כתלות באורך הייצוג n הוא

$$n = log_2 m \Rightarrow m = 2^n \Rightarrow \mathcal{O}(m) = \mathcal{O}(2^n)$$

כלומר למעשה האלגוריתם ירוץ זמן ריצה אקספוננציאלי ב<u>אורך</u> הקלט.

חשבון מודולרי

חיבור: $a + b\,(mod\,m)$ כאשר $a, b \in \mathbb{Z}_m$ ו m באורך $n \geq$ ביטים
על פי ההגדרה $0 \leq a, b < m$ ומכאן ש $0 \leq a + b < 2m$ ולכן נקבל זמן ריצה

$$\mathcal{O}(n) \quad \ni \quad \begin{cases} a + b\,(mod\,m) = a + b & \Leftarrow a + b < m \\ a + b\,(mod\,m) = a + b - m & \Leftarrow a + b \geq m \end{cases}$$

כפל: $ab\,(mod\,m)$ אשר $a, b \in \mathbb{Z}_m$ ו m באורך $n \geq$ ביטים
החישוב דורש פעולת כפל + פעולת חילוק עם שארית ובסה"כ זמן ריצה $\mathcal{O}(n^2)$

חילוק: למעשה חילוק מעל הממשיים משמעותו הכפלה באיבר ההופכי כלומר $a/b \Rightarrow ab^{-1}$ כאשר $b^{-1}b=1$
כמו שראינו בקורס באלגברה ב$\mathbb{Z}_m$ לא תמיד קיים הופכי אבל אם m ראשוני ו$0 \neq a \in$ $\mathbb{Z}_m$ אזי קיים $a' \in \mathbb{Z}_m$ כך ש $aa' = 1(mod\, m)$
במקרה כזה נאמר ש$\mathbb{Z}_m$ הוא לא רק חוג אלא שדה.

האלגוריתם של אוקלידס למציאת GCD

נעבור לדון כעת באלגוריתם $gcd(a,b)$ כאשר במהלך היום בלי הגבלת הכלליות $a \leq b$

טענה: אם $b = 0\, mod\, m$ כלומר $a \mid b$ אזי - $gcd(a,b) = a$
אחרת - $gcd(a,b) = gcd(a, a-b)$

נימוק:

$$c|a \wedge c|a \Rightarrow c|ab$$

ומנגד

$$c|a \wedge c|\left(b-a\right) \Rightarrow c|\left(a+\left(b-a\right)\right) = b$$

אם כך נוכל להמשיך טענה זו ולחסר את a מ b שוב ושוב (ועדיין להישאר עם אותו gcd) עד שנרד מתחת ל-a ומה נקבל בתוצאת החיסור הוא $b\, mod\, a$ או במילים אחרות -

מסקנה: $gcd(a,b) = gcd(a, b-ka) = gcd(a, b\, mod\, a)$

ומכאן נקבל אלגוריתם רקורסיבי:
$GCD - Euclid(a,b)$:

- $c = b\, mod\, a$
- אם $c = 0$ - נחזיר a
- אחרת - נחזיר $GCD - Euclid(c,a)$

זמן ריצה - a,b באורך $n \geq$ ביטים

בהתבוננות ראשונית נוכל לשים לב שבכל צעד אחד המספרים קטן ולכן נוכל להסיק שעומק הרקורסיה $\leq max(a,b) \leq 2^n$

ננסה לחסום באופן טוב יותר.

טענה: $b\, mod\, a \leq \dfrac{b}{a}$

הוכחה: נחלק למקרים -

1. $a \leq \frac{b}{2} \Leftarrow b\, mod\, a < a \leq \frac{b}{2}$

2. $a > \frac{b}{2}$ ⇐ כמו שראינו $b\ mod\ a = b - ka$ מאחר ו $a > \frac{b}{2}$ נקבל שאם $k > 1$

$$k > 1 \Rightarrow b\ mod\ a = b - ka \le b - 2a < 0$$

ולכן בהכרח $k = 1$ ולכן נקבל כי $b\,mod\,a = b - a < b - \frac{b}{2} = \frac{b}{2}$

אם כך בכל צעד, אחד הפרמטרים קטן לפחות בחצי ⇐כל שני צעדים, שני הפרמטרים קטנים בחצי ⇐מספר האיטרציות הוא לכל היותר $2 \cdot log_2 min(a,b)$

מאחר ו$a, b \le 2^n$ נקבל $\mathcal{O}(n)$ איטרציות, כלומר נבצע $\mathcal{O}(n)$ פעמים חילוק עם שארית ולכן סה"כ זמן ריצה $\mathcal{O}(n^3)$

משפט: אם m ראשוני ו $0 \neq a \in \mathbb{Z}_m$ אזי קיים $a' \in \mathbb{Z}_m$ כך ש $aa' = 1 \in \mathbb{Z}_m$

למה: הלמה של בזו $(Bézout)$
לכל $a, b \in \mathbb{N}$ קיים $x, y \in \mathbb{Z}$ כך ש:

$$xa + yb = GCD(a,b)$$

(נוכיח עוד מעט)

הוכחת המשפט: יהי $0 \neq a \in \mathbb{Z}_m$ כאשר m ראשוני

$$GCD(a,m) = 1 \Rightarrow \exists x, y : xa + ym = 1 \Rightarrow xa = 1 + (-y)bm \Rightarrow xa = 1(modm)$$

הערה: גם אם mפריק אבל $GCD(a,m) = 1$ קיים הופכי לa ב$\mathbb{Z}_m$

הוכחת הלמה: נשנה מעט את הלגוריתם של אוקלידס כך שיחזיר גם x, yשעבורם $xa + yb = GCD(a,b)$

$GCD - Euclid(a,b)$:

- $c = b\,mod\,a$ ונשמור את d שעבורו $b = da + c$
- אם $c = 0$: נחזיר את a וגם את $x = 1$ $y = 0$
- אחרת: נחזיר את $GCD(c,a)$ וגם את $x = y' - dx$ $y = x'$ כאשר את x', y' קיבלנו מהרקורסיה

הסבר: נניח שהקריאה הרקורסיבית החזירה x', y' כך ש

$$x'c + y'a = GCD\left(c,a\right) = GCD\left(a,b\right)$$

בפעולה $b\,mod\,a$ קיבלנו c, d כך ש

$$b = da + c$$

כלומר

$$x'c + y'a = x'(b - da) + y'a = x'b + (y' - dx')a$$

ולכן נחזיר בנוסף לGCD גם את

$$y = x' \ x = y' - dx$$

כנדרש

סבוכיות: $\mathcal{O}\left(n^3\right)$ (ניתוח זמן הריצה לא השתנה מהאלגוריתם המקורי)

בדיקת ראשוניות

חישוב חזקה בחשבון מודולרי

a, b, m באורך $n \geq$ ביטים
רוצים לחשב את $a^b \ mod \ m$

הבעיה: a^b הוא מספר באורך $ab \approx$ ביטים כלומר $\mathcal{O}\left(n2^n\right)$ ולבצע פעולות על מספרים באורך כזה זו בעיה.

רעיון: נבצע $mod \ m$ לאחר כל כפל.

זה פותר את הבעיה שהזכרנו אבל עדיין זה לא מספיק משום שאנחנו נדרשים לבצע b פעולות/איטרציות כאשר $b = \mathcal{O}\left(2^n\right)$ כלומר נקבל זמן ריצה אקספוננציאלי באורך הקלט n.

טריק נפוץ ושימושי: נחשב את הסדרה

$$a \, mod \, m, \ a^2 \, mod \, m, \ a^4 \, mod \, m, \ \ldots, a^{2^n} \ mod \, m$$

סדרה בת n איברים
בשביל לחשב כל איבר בסדרה פשוט נעלה את קודמו בריבוע

$$\left(a^i \ mod \ m\right)^2 = a^{2i} \ mod \ m$$

נשים לב שמתכונות החשבון המודולורי תוצאת ה mod תהיה זהה גם אם נבצע אותה אחרי כל העלאה בריבוע, וכך נמנע מלבצע פעולות חשבוניות עם מספרים גדולים מדי.
נקבל שלחישוב כל איבר נזדקק לפעולת כפל + פעולת mod (ששוות ערך לחילוק עם שארית) כלומר $\mathcal{O}\left(n^2\right)$ פעולות
ולכן בסך הכל עבור ככל הסדרה נקבל שזמן החישוב הוא $\mathcal{O}\left(n^3\right)$
כעת נוכל לפרק את החזקה (בעזרת הייצוג הבינארי שלה) לסכום של חזקות של 2 כלומר

$$b = 2^{x_1} + 2^{x_2} + \ldots$$

נקבל, אם כך, שנוכל לחשב את פעולת העלאה בחזקה בעזרת חישוב כפל של חזקות

$$a^b = a^{2^{x_1}} \cdot a^{2^{x_2}} \ldots$$

גם כאן נכניס את פעולת ה mod פנימה ונקבל:

$$b = \sum 2^{x_k} \Rightarrow a^b \ mod \ m = \prod_k (a^{2^{x_k}} \ mod \ m)$$

דוגמה:

$$5^{13}\ mod\ 7$$

$$5\,mod\,7, \Rightarrow\ 5^2 = 25\,mod\,7 = 4 \Rightarrow\ 5^4 = 4^2 = 2\,mod\,7 \Rightarrow 5^8 = 2^2 = 4\,mod\,7$$

$$5^{13}\ mod\ 7 = 5^{\sum\left(2^3+2^2+2^0\right)}\ mod\ 7 = 5^8 \cdot 5^4 \cdot 5^1\ mod\ 7 = 4 \cdot 2 \cdot 5 = 5\ mod\ 7$$

זמן ריצה:

- עיבוד ראשוני - $\mathcal{O}\left(n^3\right)$ עבור חישוב הסדרה
- בכל שלב עבור כל 2^{x_k} נבצע -

$$(\prod_{j=1}^{k-1} a^{2^{x_j}}\ mod\,m)(a^{2^{x_k}}\ mod\,m)\ mod\,m$$

כאשר הכופל השמאלי הוא מה שחושב עד כה והימני הוא החישוב הבא המבוקש

- ולכן מדובר בכל שלב בכפל + mod $\Leftarrow$ סה"כ $\mathcal{O}\left(n^3\right)$

$PRIMES$

ישנן שתי בעיות קרובות אבל שונות מאוד בתחום של ראשוניות:

1. בדיקת ראשוניות $(PRIMES)$ - בהינתן $m \in \mathbb{N}$ האם הוא ראשוני?
2. פירוק לגורמים $(FACTORING)$ - בהינתן $m \in \mathbb{N}$ פריק, מצא גורם של m
   בהנתן אלגוריתם שפותר את בעיה 2 נוכל למצוא בעזרתו את כל הגורמים של m - נחלק במה שמצאנו ונפעיל את האלגוריתם על תוצאת החילוק.

בעיה 1 היא בעיה קלה - שיערו שזה כך, ואכן בשנת 2002 הוכח שהיא בP

בעיה 2 לעומת זאת נחשבת בעיה קשה - לא ידוע על אלגוריתם שפותר אותה בזמן סביר.

1.11.15 היסטוריה

1976: אלגוריתם של $Miller$ דטרמיניסטי ופולינומי
$Miller$ הוכיח שהאלגוריתם נכון אם השערת רימן המוכללת נכונה

1977: אלגוריתם של $Solovag-Starassen$ רנדומי, פולינומי, נכון ללא הנחות
הם הראו ש $PRIMES \in CO-RP$
$CO-RP$ - אלגוריתמים בעלי אפשרות רנדומית לטעות חד־צדדית.
כלומר במקרה שלנו - אם m ראשוני - האלגוריתם יחזיר "כן" תמיד, אם הוא פריק - האלגוריתם יחזיר "לא" בהסתברות $\leq \frac{1}{2}$
אם חוזרים k פעמים על האלגוריתם על m נתון, ההסתברות שנחזיר <u>בכל</u> הפעמים "כן" כאשר למעשה m פריק היא $\leq \left(\frac{1}{2}\right)^k$

1980: $Miller-Rabin$ אלגוריתם ב $CO-RP$ (כלומר עם שגיאה הסתברותית חד צדדית), יעיל יותר (דרגת הפולינום נמוכה יותר), נכון ללא הנחות

1992: $Adelman-Hung$ אלגוריתם ZPP כלומר רנדומי במובן הזה שהוא תמיד מחזיר תשובה נכונה ותוחלת זמן הריצה היא פולינומית

הערה: נניח שתוחלת זמן הריצה n^c נקבל מאי־שיוויון מרקוב $Pr\left[run-time\geq 2n^c\right]\leq \frac{1}{2}$ ולכן אם נריץ את האלגוריתם k פעמים כל פעם במשך $2n^c$ צעדים ההסתברות שאף ריצה לא תעצור בזמן הזה היא $\leq\left(\frac{1}{2}\right)^k$

2001: $Agrawal,\ Kayal,\ Saxena$ הוכיחו ש $PRIMES\in P$

מציאת מספר ראשוני

הרעיון הכללי: נגריל מספר באורך m ביטים ונריץ אלגוריתם לבדיקת ראשוניות, אם התשובה היא "כן" נחזיר את m.

משפט המספרים הראשוניים: נגדיר

$$\pi(x)=\left|\{p|p\leq x,\ p\,is\,prime\}\right|$$

אזי

$$\pi(x)=(1+o(1))\frac{x}{ln\,(x)}$$

ומכאן שבין x ל$2x$ יש $(1+o(1))\frac{x}{ln(x)}$ ראשוניים

כלומר אם נגריל מספר שלם $2^n\leq m\leq 2^{n+1}$ נקבל ש

$$Pr\left[m\ is\ prime\right]=\frac{(1+o(1))\frac{2^n}{ln(2^n)}}{2^n}=(1+o(1))\frac{1}{nln\,(2)}$$

אם נבצע kn חזרות ההסתברות להצלחה באחת מהן היא

$$1-Pr\left[failor\right]=1-\left(1-\frac{1+o\,(1)}{nln2}\right)^{kn}\geq 1-\left(e^{-\frac{1+o(1)}{nln2}}\right)^{kn}=1-\left(e^{-\frac{1+o(1)}{ln2}}\right)^{k}>1-\left(\frac{1}{4}\right)^{k}$$

השערה: השערת $Cramer$ - הפער המקסימלי בין שני ראשונים עוקבים באורך n בייים הוא $\mathcal{O}\left(n^2\right)$

התוצאה הכי טובה בדרך להוכחת ההשערה היא שהפער לא גדול מ $\approx\frac{1}{n}2^{\frac{n}{2}}$

אלגוריתם $CO-RP$ לבדיקת ראשוניות

עד פשוט לפריקות: m פריק $\Leftarrow$ קיים $0<a<m$ כך ש $GCD(a,m)\neq 1$

דוגמה: אם $m=pq$ כאשר p,q ראשוניים באורך n $\left(2^{2n}\approx m,\ 2^n\approx p,q\right)$ כמה עדים יהיו?

$$\begin{cases}\text{p,2p,...}(q-1)p\\ q,2q,...,(p-1)q\end{cases}\Rightarrow p+q+2\ \text{עדים מתוך}\ m$$

כלומר אם נגריל מספר, ההסתברות שנפגע בעד היא בערך $\mathcal{O}\left(\frac{2\cdot 2^n}{2^{2n}}\right)=\mathcal{O}\left(\frac{1}{2^n}\right)$ - לא יעיל!

משפט: משפט פרמה הקטן
אם p ראשוני אזי לכל $1 < a < p-1$ מתקיים $a^{p-1} \equiv 1 \, mod \, p$

הוכחה: נבחר $1 < a < p-1$ ונסתכל על הקבוצה

$$A = \{a \cdot i \, mod \, p \mid i = 1...p-1\}$$

$\mathbb{Z}_p$ שדה $\Leftarrow$ אם $i, j \in \mathbb{Z}_p$ כך ש

$$i \equiv j \, mod \, p \Leftarrow (i-j)a\bar{a} \equiv 0 \, mod \, p \Leftarrow (i-j)a \equiv 0 \Leftarrow ia \equiv ja$$

מנימוק דומה ניתן להראות שכל אברי הקבוצה שונים מ0 ולכן למעשה אברי A הם כל המספרים $1, ..., p-1$ בפרמוטציה כלשהי
ומכאן

$$0 \overset{*}{\not\equiv} \prod_{i=1}^{p-1} i \equiv \prod_{i=1}^{p-1} ai \equiv a^{p-1} \prod_{i=1}^{p-1} i \;\; (mod \, p)$$

השיוויון * נובע מכך שבשדה מכפלה של אברים שאינם אפס בהכרח שונה מאפס.
מאחר ואנחנו בשדה לכל איבר קיים הופכי ולכן נוכל להכפיל ב $\left(\prod_{i=1}^{p-1} i\right)^{-1}$ ולקבל

$$a^{p-1} \equiv \left(\prod_{i=1}^{p-1} i\right)\left(\prod_{i=1}^{p-1} i\right)^{-1} \equiv 1 \;\; (mod \, p)$$

ציינו מקודם שאם m פריק קיים $0 < a < m$ כך ש $GCD(a,m) \neq 1$ ובפרט

$$a^{m-1} \neq 1 \, mod \, m$$

כי אם $c|a, m$ נקבל

$$\forall j, k: \;\; c|a^k - jm \Rightarrow c|a^k \; mod \; m$$

כלומר מחזקות של a נקבל תמיד מספר שמחלק את c (מודולו m) ולא נקבל 1

שאלה: בעזרת הטענה אפשר לשלול ראשוניות באופן חד משמעי (אם היא לא מתקיימת עבור a כלשהו) אבל מה יקרה אם m למעשה פריק? אז יתכן מצב שבו קיים a שיקיים את השקילות $a^{m-1} \equiv 1 \; mod \; m$. נרצה לוודא שאם המספר פריק בדיקת השקילות של משפט פרמה תכשל בסבירות גבוה.
אם כך נשאל - מה קורה אם קיים a כך ש

$$GCD(a,m) = 1$$

ובנוסף a הוא עד לפריקות על פי פרמה כלומר

$$a^{m-1} \not\equiv 1 \, mod \, m$$

או במילים אחרות a אמנם זר ל m ולכן לא יכול להפריד את הראשוניות של m באופן הנאיבי שהצענו אבל מצד שני הוא כן מפריד את הראשוניות על פי פרמה?

למה: אם קיים a כך ש $GCD(a,m)=1$ ובנוסף $a^{m-1}\not\equiv 1\,mod\,m$
אזי לפחות חצי מהמספרים $b\in\{1,..,m-1\}$ מקיימים גם $b^{m-1}\not\equiv 1\,mod\,m$

הוכחה: נניח שקיים a כזה ונגדיר

$$X=\left\{1\leq x\leq m-1|x^{m-1}\not\equiv 1\,mod\,m\right\}$$

נסמן את שאר האיברים בטווח

$$Y=\left\{1\leq y\leq m-1|y^{m-1}\equiv 1\,mod\,m\right\}$$

נראה ש $|Y|<|X|$ על ידי המיפוי החד־חד־ערכי מ Y ל־X הבא:

$$y\in Y\mapsto ay\,mod\,m$$

נראה תחילה שזו אכן מעתיקה איברים מ Y ל־ X

$$(ay)^{m-1}\equiv a^{m-1}y^{m-1}\equiv a^{m-1}\not\equiv 1\,mod\,m\Rightarrow ay\in X$$

הראנו בעבר שגם אם $\mathbb{Z}_m$לא שדה עבור a כך ש $GCD(a,m)=1$ קיים הופכי ב Z_m
נשתמש בעובדה זו כדי להראות את החד־חד־ערכיות של ההעתקה שהגדרנו

$$ay\equiv az\,mod\,m\Rightarrow a^{-1}ay\equiv a^{-1}az\,mod\,m\Rightarrow y\equiv z\,mod\,m$$

מסקנה: אם קיים a שהינו עד שסותר את משפט פרמה הקטן אבל הוא זק ל־m אזי יש "הרבה" כאלה (יותר מ$\frac{1}{2}$) עדים כאלה.
ולכן נוכל להגדיר את האלגוריתם הבא:

$: Not-Quite-Miller-Rabin(m)$

- נגריל $a\in\{1,...,m-1\}$ ונבדוק
  - אם $a^{m-1}\equiv 1\,mod\,m$: נחזיר "כן"
  - אחרת : נחזיר "לא"

אם m ראשוני אז על פי משפט פרמה הקטן תנאי הבדיקה יהיה תמיד חיובי ולכן תמיד נזהה נכון ונחזיר "כן"

אם m פריק אז ־ או שתנאי הבדיקה של משפט פרמה הקטן יכשל ונזהה נכון את m כפריק או שבמקרה ניפול על $a\in Y$ כלומר $a^{m-1}\equiv 1\,mod\,m$ ואז נטעה ונחשוב ש m ראשוני. אבל הראנו שהסיכוי שהאפשרות השניה תקרה היא קטנה מ $\frac{1}{2}$ כלומר במקרה ש m פריק נחזיר תשובה נכונה בהסתברות $\leq\frac{1}{2}$.

5.11.15 **הגדרה:** מספרי קרמייקל $Carmichael$
מספר $m\in\mathbb{N}$ כך שלכל a כך ש $GCD(a,m)=1$ ומתקיים $a^{m-1}\equiv 1\,mod\,m$
כלומר זהו מספר שאין עבורו עדים מהסוג של משפט פרמה הקטן ועבורתם האלגוריתם שתיארנו יכשל בסבירות 1 (ולא $\frac{1}{2}$ כפי שרצינו).

בעיה: יש אינסוף מספרי קרמייקלץ למרות שהם נדירים

משפט: עבור n ביטים

$$Pr\left[m\ is\ Carmichael\ number\right] \leq e^{-\Omega\left(n^{\frac{log(log(n))}{log(n)}}\right)}$$

לעומת זאת

$$Pr\left[m\ is\ prime\right] = \Theta\left(\frac{1}{n}\right)$$

ולכן האלגוריתם שראינו מספיק טוב כדי להגריל ולזהות מספר ראשוני בהסתברות גבוה מאוד. אם נפלנו על ראשוני אז מצוין. אם נפלנו על סתם מספר פריק בהרבה הרצות של הבדיקה נקבל סיכוי נמוך מאוד שנטעה ונחשוב שהוא ראשוני והסיכוי שכל הבדיקה נכשלה כי נפלנו על מספר קרמייקל הוא גם קלוש כי הם ממש נדירים.

אבל בתור אלגוריתם לבדיקה של מספר נתון זה לא מספיק, כי כאשר כבר נתון מספר לא מעניינת אותנו ההסתברות לקבל דווקא אותו ואם נפלנו על אחד בעייתי ניכשל בוודאות בלי קשר לכמה פעמים נבדוק. לכן הוסיפו באלגוריתם שלב של בדיקה שמזהה מספרי קרמייקל.

אבחנה: אם p ראשוני

ומתקיים $x^2 \equiv 1\, mod\, p$ אזי:

$$(x+1)(x-1) = x^2 - 1 \equiv 0\, mod\, p$$

ומראשוניות p

$$p|(x+1)(x-1) \Rightarrow p|x+1\ or\ p|x-1 \Rightarrow x \equiv \pm 1\, mod\, m$$

ולכן עד נוסף לפריקות m : $a < m$ כך ש $a^2 \equiv 1\, mod\, m$ אבל $a \not\equiv \pm 1\, mod\, m$

וכעת

$Miller - Rabin(m)$

- נגריל באופן אחיד $a \in \{1, ..., m-1\}$
- אם $a^{m-1} \not\equiv 1\, mod\, m$: נחזיר "לא"
- אחרת (כלומר $a^{m-1} \equiv 1\, mod\, m$)
  - נכתוב $m - 1 = 2^t q$ עבור $t \in \mathbb{N}$ ו q אי־זוגי
  - נחשב את הסדרה

$$a_0 = a^q\, mod\, m, a_1 = a^{2q}\, mod\, m, ..., a_t = a^{2^t q}\, mod\, m = a^{m-1}\, mod\, m = 1$$

  - לכל הסדרה: אם קיים $j \in \{1, ..., t\}$ כך ש $a_j = 1$ ו $a_{j-1} \neq \pm 1$: נחזיר "לא"

– אחרת : נחזיר "כן"

הסבר לצעד הנוסף: אם קיים j כמו שמתואר בצעד כלומר קיים

$$a_j = a^{2^j q} = \left(a^{2^{j-1} q}\right)^2 = 1 \ mod \ m$$

ובנוסף

$$a^{2^{t-1} q} = a_{j-1} \neq 1 \ mod \ m$$

ולכן a_{j-1} הוא עד לפריקות כפי שהראנו מקודם.

משפט: אם m מספר קרמייקל אזי הבדיקה הנוספת תחזיר "לא" בהסתברות $\frac{3}{4}$ (ללא הוכחה)

זמן ריצה:

- חישוב $a^m \ mod \ m$: $\mathcal{O}\left(n^3\right)$
- לחשב $a_0 = a^q \ mod \ m$: $\mathcal{O}\left(n^3\right)$
- חישוב $a_j = a_{j-1} \cdot a_{j-1} \ mod \ m$: $\mathcal{O}\left(n^2\right)$ ונבצע זאת מספר פעמים : $t \leq log\left(m\right) = \mathcal{O}\left(n\right)$

ולכן סב"כ - $\mathcal{O}\left(n^3\right)$

קריפטוגרפיה

הצפנה במובן הקלאסי דורשת מפתח שבעזרתו ניתן להצפין הודעות ולפענח אותם. הצפנה שכזאת מכונה - הצפנה סימטרית.

בשנת 1977 פרסמו $Diffie, Hellman$ מאמר ובו העלו את הרעיון שניתן לבנות מערכות הצפנה שאינן סימטריותץ הם אף תיארו פרוטוקול ראשוני בעל אופי לא סימטרי.

סכימה כללית של פרוטוקול הצפנה במפתח ציבורי:

1. בוריס מייצר (לפי אלגוריתם אקראי) את המפתחות (e, d) כאשר $d = private \ key, \ e = public \ key$
2. בוריס מפרסם את e ושומר אצלו את d
3. אנסטסיה מצפינה את המסר x באמצעות $E\left(x, e\right) = y$
4. בוריס מפענח את המסר y באמצעות $D\left(y, d\right) = x$

הנחת הקושי: אי אפשר לגלות את x בהסתברות סבירה בלי d

הנחה יותר פורמלית (ויותר מחמירה): לכל אלגוריתם A ולכל שתי הודעות x_1, x_2 מתקיים[1]

$$Pr\left[A\left(E\left(x_1, e\right), e\right) = 1\right] \approx Pr\left[A\left(E\left(x_2, e\right), e\right) = 1\right]$$

ייצור המפתחות:

- בוריס מגריל באקראי שני מספרים ראשוניים גדולים - p, q ומחשב $N = p \cdot q$
- בוחר e כך ש $GCD\left(e, (p-1)(q-1)\right) = 1$
- מחשב באמצעות האלגוריתם של אוקלידס את d כך ש $ed = 1\ mod\ \ (p-1)(q-1)$
- מפרסם את (N, e)
- שומר לעצמו את (d)

אנסטסיה רוצה לשלוח לבוריס את ההודעה x:

- מצפינה את x באופן הבא $y = x^e\ mod\ N$
- שולחת לבוריס את y

בוריס רוצה לפענח:

- מחשב את $y^d\ mod\ N$

טענה: $y^d\ mod\ N = x$

הוכחה קצרה ולא סחומר: גודל החבורה $\mathbb{Z}_N^*$ (חבורת המספרים הזרים לN) היא $\varphi(N) = (p-1)(q-1)$ ועל פי משפט מתורת החבורות

$$\forall a, b \in \mathbb{Z}_N^* :\ \ a^b \equiv a^{b\ mod\ \varphi(N)}\ mod\ N$$

ולכן מאחר ו $ed = 1\ mod\ \varphi(N)$ נקבל

$$y^d = (x^e)^d = x^{ed} \equiv x^1 \equiv x\ mod\ N$$

הוכחה קצרה פחות וכן בחומר: לפי המשפט הקטן של פרמה

$$x^{p-1} = 1\ mod\ p,\ \ x^{q-1} = 1\ mod\ q$$

ולכן

$$ed = 1\ mod\ \ (p-1)(q-1) \Rightarrow ed = 1 + c(p-1)(q-1)$$

$$\Rightarrow y^d = (x^e)^d = x^{ed} = x^{1+c(p-1)(q-1)} = x\left(x^{p-1}\right)^{c(q-1)}$$

אבל $x^{p-1} = 1\ mod\ p$ ולכן

$$\Rightarrow x^{ed} = x \cdot 1^{c(q-1)}\ mod\ p = x\ mod\ p$$

---

[1]למעשה יש הגדרה עוד יותר פורמלית שמגדירה במדויק מה הכוונה $\approx$

באותו אופן נקבל

$$x^{ed} = x \ mod \ q$$

ומכאן נקבל ש $x^{ed} - x$ מתחלק ב־p וגם ב־q ומאחר והם ראשוניים הוא מתחלק גם במכפלה $pq = N$ ולכן בסה"כ

$$x^{ed} - x = 0 \ mod \ N \Rightarrow x^{ed} = x \ mod \ N \qquad \blacksquare$$

אם יבגני (שמצוטט לקו ומנסה להבין מה המסר שעבר מאנסטסיה לבוריס) ידע לפרק את N אזי הוא יוכל לעשות את אותם חישובים בדיוק כמו בוריס ולפענח את המסר המוצפן.

הנחת קושי: בהינתן y, e, N קשה לחשב את x בלי לדעת את d (לא קיימת הוכחה[2])

הפרד ומשול

כפל מספרים

נרצה לנסות לחסוך בפעולות הדרושות לשם חישוב כפל.

נתבונן אם כך בשני מספרים בינארים מאורך 2^n - a, b

נחלק כל אחד מהם לשרשור של שני חלקים שווים:

$$a = a_1 a_2 = \overbrace{...a_1...}^{n/2 \ bits} \overbrace{...a_2...}^{n/2 \ bits} = a_1 \cdot 2^{\frac{n}{2}} + a_2$$

$$b = b_1 b_2 = \overbrace{...b_1...}^{n/2 \ bits} \overbrace{...b_2...}^{n/2 \ bits} = b_1 \cdot 2^{\frac{n}{2}} + b_2$$

והכפל ביניהם יתן

$$a \cdot b = a_1 b_1 2^n + (a_1 b_2 + a_2 b_1) \, 2^{\frac{n}{2}} + a_2 b_2$$

נשים לב שהכפלה ב2^k פירושה הזזה של תוצאת הכפל בk ביטים. פעולה לא משמעותית מבחינת זמן הריצה.

הרעיון הוא לנסות לבצע ברקורסיה את הכפל בין החצאים השונים. אלא שבמצב הנוכחי בכל שלב ברקורסיה נבצע 4 קריאות רקורסיביות

$$1. \ a_1 b_1 \ 2. \ a_1 b_2 \ 3. \ a_2 b_1 \ 4. \ a_2 b_2$$

ונקבל בדיוק את אותו זמן ריצה כמו באלגוריתם הנאיבי שאנו מכירים (נראה את חישוב זמן הריצה בהמשך).

אבל נשים לב שמתקיים

$$a_1 b_2 + a_2 b_1 = (a_1 + a_2) \, (b_1 + b_2) - a_1 b_1 - a_2 b_2$$

ואת $a_1 b_2$ ואת $a_2 b_1$ אנו מחשבים ממילא ולכן נוכל בעזרת השיווין הזה לחסוך קריאה רקורסיבית אחת.

---

[2] בתרגיל בית ראינו את "הצפנת רבין" ועבורה הראנו שקילות לבעיית הפירוק לגורמים

אלגוריתם קרצובה $Karatsuba\left(a,b,n\right)$:

- אם $n=1$: נחזיר את $a\cdot b$
- אחרת:
  – נחלק את a,b ל a_1,a_2,b_1,b_2 כמו שתיארנו למעלה (מדובר בסך הכל בכמה פעולות הזזה)
  – נחשבת רקורסיבית :

$$k_1\leftarrow Karatsuba\left(a_1,b_1,\frac{n}{2}\right)$$

$$k_2\leftarrow Karatsuba\left(a_2,b_2,\frac{n}{2}\right)$$

$$k_3\leftarrow Karatsuba\left(\left(a_1+a_2\right),\left(b_1+b_2\right),\frac{n}{2}\right)$$

  – ונחזיר

$$k_2+2^nk_1+2^{\frac{n}{2}}\left(k_3-k_2-k_1\right)$$

זמן ריצה:

קריאה אחת בלי רקורסיה : $\mathcal{O}\left(n\right)$

עבור הקריאות הרקורסיביות:

נסמן ב $T\left(n\right)$את זמן הריצה עבור מספר באורך n

נקבל

$$T\left(n\right)=3T\left(\frac{n}{2}\right)+\mathcal{O}\left(n\right)$$

בכל שלב ברקורסיה נקרא ל3 קריאות. מאחר ובכל קריאה אנחנו מחלקסי את n ב2 עומק הרקורסיה יהיה $log_2\left(n\right)$

לא נחשב במדויק (ראינו בעבר שיטות איך לעשות זאת) אלא נתאר באופן כללי עץ קריאות רקורסיביות. לכל קודקוד בעץ יהיו 3 בנים ועומק העץ יהיה $log_2\left(n\right)$ ולכן מספר העלים יהיה

$$3^{log_2(n)}=n^{log_23}\approx n^{1.584}$$

זמן הריצה שכל קודקוד מתאר הוא למעשה סכום של הבנים שלו ועוד זמן לינארי שלא משפיע על החישוב. לכן נקבל שזמן הריצה הסופי שווה אסימפטוטית למספר העליסץ כלומר

$$T\left(n\right)=\mathcal{O}\left(n^{log_23}\right)\approx\mathcal{O}\left(n^{1.584}\right)$$

שזה שיפור לעומת ה $\mathcal{O}\left(n^2\right)$[3]

מאז נעשו עוד שיפורים בזמן הריצה. האלגוריתם הכי יעיל שידוע כיום הוא של $F\ddot{u}rer-2007$ שרץ בזמן - $\mathcal{O}\left(n\cdot log\left(n\right)\cdot 2^{\Theta\left(log^*(n)\right)}\right)$

[3]נשים לב שאם לא היינו מצמצמים אלא נשארים עם 4 קריאות רקורסיביות בכל שלב היינו מקבלים זמן ריצה $\mathcal{O}\left(4^{log_2n}\right)=\mathcal{O}\left(n^{log_24}\right)=\mathcal{O}\left(n^2\right)$ כלומר זהה לאלגוריתם הנאיבי.

מכפלת מטריצות

יהיו A,B מטריצות nxn נרצה לייעל את זמן הריצה של פעולת ההכפלה ביניהם.

יש n^2 תוצאות שצריך לחשב ולכן זמן הריצה יהיה לכל הפחות n^2.

אלגוריתם נאיבי - לכל תא במטריצה נבצע את הכפלת השורה והעמודה המתאימות כלומר נבצע $\mathcal{O}\left(n\right)$ פעולות כאורך עמודה/שורה. בסה"כ נקבל $\mathcal{O}\left(n^3\right)$ עבור כל החישוב.

ננסה לצמצם את מספר הפעולות באופן הדומה לזה שראינו באלגוריתם קרצובה.

יהיו X,Y נחלק אותם ל4 מטריצות בלוקים כל אחד בגודל $n/2xn/2$

$$X=\begin{pmatrix}A & B\\ C & D\end{pmatrix},\ Y=\begin{pmatrix}E & F\\ G & H\end{pmatrix}$$

תוצאת הכפל תהיה בייצוג הזה

$$XY=\begin{pmatrix}AE+BG & AF+BH\\ CE+DG & CF+DH\end{pmatrix}$$

אם ננסה כעת לחשב ברקורסיה את כל המכפלות נקבל 8 קריאות רקורסיביות בדומה לחישוב שראינו לגבי כפל מספרים נקבל זמן ריצה

$$\mathcal{O}\left(8^{log_2 n}\right)=\mathcal{O}\left(n^{log_2 8}\right)=\mathcal{O}\left(n^3\right)$$

כמו באלגוריתם הנאיבי. משום כך ננסה לצמצם את מספר הקריאות, אפילו הורדה של קריאה אחת כבר תהווה שיפור בזמן הריצה.

אלגוריתם שטראסן $Strassen-1969$**:**

•

$$P_1=A\left(F-H\right),P_2=\left(A+B\right)H,P_3=\left(C+D\right)E,P_4=D\left(G+E\right),$$

$$P_5=\left(A+D\right)\left(E+H\right),P_6=\left(B-D\right)\left(G+H\right),P_7=\left(A-C\right)\left(E+F\right)$$

• והתוצאה תתקבל על ידי

$$XY=\begin{pmatrix}P_4+P_5+P_6-P_2 & P_1+P_2\\ P_3+P_4 & P_4+P_5-P_7\end{pmatrix}$$

זמן ריצה: אחרי האלגוריתם של שטראסן התקבלו הרבה מאוד תוצאות ושיפורים וצמח תחום שלם של אלגוריתמים לחישוב כפל מטריצות. ובעקבות זאת החליטו לתת סימון מיוחד כדי לסמן את זמני הריצה של אלגוריתמים בתחום - ω.

האלגוריתם שראינו עכשיו נותן $\omega=log_2 7$

בשנים שאחרי התקבלו התוצאות הבאות

$$\omega=2.796,2.78,2.548,2.5222,2.517,2.416,2.409,2.376$$

התוצאה האחרונה ברשימה התקבלה בסוף שנות ה80 ומאז במשך שנים אף אחד לא הצליח לשפר. לפני 4 שנים $Wiliams$ הצליחה להשיג $\omega=2.3727$ [התברר אחרי זה שמאסטרנט בשם $Stathers$ הצליח כמה חודשים לפני להשיג $\omega=2.3275$ אבל אף אחד לא שמע על זה כי הוא לא טרח לפרסם את זה כמו שצריך]

12.11.15 **כפל פולינומים והתמרת פורייה**

נתונים שני פולינומים ממשיים נדרגה $n\geq$

$$a\left(x\right)=\sum_{i=0}^{n}a_{i}x^{i},\ b\left(x\right)=\sum_{i=0}^{n}b_{i}x^{i}$$

הייצוג של הפולינומים, שהוא למעשה הנתון שלנו, יהיה, בשלב זה, על ידי סדרת המקדמים של הפולינום. כלומר נתונות לנו שתי סדרות של מקדמים מששיים.

רוצים למצוא את המכפלה שלהם

$$a\left(x\right)b\left(x\right)=c\left(x\right)=\sum_{i=0}^{n}c_{i}x^{i}$$

כלומר רוצים למצוא את סדרת המקדמים $\left\{ c_{0},...,c_{n}\right\}$ כך ש

$$c_{k}=\sum_{j=0}^{k}a_{j}b_{k-j}$$

אם נחשב כל מקדם באופן הנאיבי הזה, עבור כל מקדם c_{k} נבצע $\mathcal{O}\left(k\right)$ פעולות כפל. מאחר ויש $\mathcal{O}\left(n\right)$ מקדמים נקבל בסה"כ

$$1+2+...+n=\mathcal{O}\left(n^{2}\right)$$

נשים לב לפער בין מספר פעולות הכפל שקיבלנו לבין אורך ה<u>פלט</u> (באופן כללי אורך הפלט מהווה חסם תחתון לזמן הריצה, שהרי המינימום אותו יש לעשות הוא להדפיס את הפלט. הרבה פעמים לא ניתן להגיע ממש עד לחסם התחתון הזה אבל ננסה כמה שניתן לצמצם את הפער עד אליו).

אנו נראה איך ניתן לשפר את התוצאה הזאת עד כדי $\mathcal{O}\left(n\cdot log\left(n\right)\right)$ בעזרת מושג שנקרא "התמרת פורייה".

כדי להתעסק בנושא נתחיל בתזכורת/מבוא על פונקציות מרוכבות:

פונקציות מרוכבות - על רגל אחת

אנו מתעסקים במרחב המספרים המרוכבים

$$\mathbb{C}=\left\{ a+bi\mid a,b\in\mathbb{R},i=\sqrt{-1}\right\} to$$

נוסחת אוילר אומרת ש־

$$e^{i\pi}-1=0$$

למה זה נכון? נתבונן בטור טיילור של פונקציית האקספוננט

$$e^{x}=\sum_{n=0}^{\infty}\frac{x^{n}}{n!}$$

נזכור שמתקיים

$$i^{0}=1,\ i^{1}=i,\ i^{2}=-1,$$

$$i^{3}=-1\cdot i=-i,\ i^{4}=-i\cdot i=-\left(-1\right)=1$$

ולכן אם נציב בפונקציה xi נקבל

$$e^{xi}=\frac{\left(xi\right)^{0}}{1}+\frac{\left(xi\right)^{1}}{1}+\frac{\left(xi\right)^{2}}{2}+\frac{\left(xi\right)^{3}}{6}+\frac{\left(xi\right)^{4}}{24}...$$

$$=\frac{x^{0}}{1}1+\frac{x^{1}}{1}i+\frac{x^{2}}{2}\left(-1\right)+\frac{x^{3}}{6}\left(-i\right)+\frac{x^{4}}{24}1...$$

$$=\frac{x^{0}}{1}+\frac{x^{1}}{1}i-\frac{x^{2}}{2}-\frac{x^{3}}{6}i+\frac{x^{4}}{24}...$$

נפריד את הסכום לשניים ־ האיברים שמוכפלים בi ואלה שלא

$$e^{xi}=\begin{cases}\frac{1}{1}-\frac{x^{2}}{2}+\frac{x^{4}}{4!}+...\\ +\left(\frac{x}{1}-\frac{x^{3}}{3!}+\frac{x^{5}}{5!}+...\right) & i\end{cases}$$

ואלה למעשה טורי טיילור של פונקציות $cos\left(x\right)$ ו $sin\left(x\right)$ ולכן

$$e^{xi}=cos\left(x\right)+i\cdot sin\left(x\right)$$

או בסימון מקוצר

$$e^{xi}=cis\left(x\right)$$

כעת אם נציב $x=\pi$ נקבל

$$e^{\pi i}=cos\left(\pi\right)+i\cdot sin\left(\pi\right)=-1+i\cdot0=-1$$

באופן מקביל אפשר להתייחס אל $r\cdot e^{\theta i}$ בתור דרך הצגה אחרת של מספרים מרוכבים.

אפשר לראות כל מספר מרוכב כנקודה במישור המרוכב הדו־מימדי. לכל נקודה כזאת נתבונן בישר ממנה לראשית הצירים (המספר המרוכב $0+0i$) ונסמן ב θ את הזווית מהישר לציר x וב r את האורך של הישר. במילים אחתרחקים מהראשית על הציר הממשי ("ציר x") לפי r ואז "מסתובבים" בזווית θ. הצגה זו $\left(r,\theta\right)$ נקראית ־ "הצגה פולרית" או "הצגה קובטית".

אם נעשה את החשבון (לא נעשה אותו כעת) נוכל לראות שהמעבר מהצגה זו להצגה ה"רגילה" של $a+bi$ ממירה נקודה בייצוג פולרי (r,θ) לנקודה

$$r\left(cos\left(\theta\right)+i\cdot sin\left(\theta\right)\right)=r\cdot e^{\theta i}$$

אם נתיחחס למקרה הפרטי בו $r=1$ נקבל ש e^{xi} מייצג למעשה נקודות על מעגל היחידה (המרחק r מהראשית הוא 1, הזווית היא המשתנה).

הבצגה הזאת כפל של מספרים מרוכבים נעשה פשוט וברור יותר

$$z_1=r_1\cdot cis\left(\theta_1\right),\ \ z_2=r_2\cdot cis\left(\theta_2\right)$$

$$z_1\cdot z_2=r_1\cdot e^{\theta_1 i}\cdot r_2\cdot e^{\theta_2 i}=\left(r_1r_2\right)e^{\left(\theta_1+\theta_2\right)i}=\left(r_1r_2\right)cis\left(\theta_1+\theta_2\right)$$

ובאותו אופן נקבל שהעלאה בריבוע של $e^{\theta i}$ תיתן

$$e^{\theta i}\cdot e^{\theta i}=e^{2\theta i}$$

אם נחזור לפרשנות הגיאומטרית שהכזרנו למעלה, המשמעות של פעולת העלאה בריבוע היא סיבוב של נקודה על מעגל היחידה, זווית הסיבוב היא θ.[4]

באופן כללי נוכל להראות באינדוקציה שהעלאה בחזקה היא

$$\left(e^{\theta i}\right)^n=e^{n\theta i}$$

הערה: ההצגה לא יחידה שהרי

$$r\cdot e^{\theta i}=r\cdot e^{(2\pi k+\theta)i}$$

עבור חזקות שלמות (כלומר $\left(r\cdot e^{\theta i}\right)^n$ כאשר $n\in\mathbb{Z}$) נקבל ש

$$\left(r\cdot e^{\theta i}\right)^n=r^n\cdot e^{n\theta i}=r^n\cdot e^{2\pi nk+n\theta i}=\left(r\cdot e^{2\pi k+\theta i}\right)^n$$

ולכן יש לנו סוג של "סגירות" תחת $+2\pi k$ ולכן חוסר היחידות לא באמת מהווה בעיה.

אבל עבור חזקות לא שלמות נקבל שיש לנו בעיה של הגדרה, למעשה ההצגה הזאת לא לגמרי מוגדרת היטב.

ערך מוחלט מוגדר כ[5]

$$|z|=\begin{cases}|a+bi| & =\sqrt{a^2+b^2}=\sqrt{z\cdot\bar{z}}\\ |r\cdot cis\left(\theta\right)| & =r\end{cases}$$

אפשר לומר שהערך המוחלט מודד את המרחק מהמספר 0 ולכן, בהתאם להסבר הגיאומטרי לעיל, מתבקש שאכן בהצגה הזאת נקבל שהוא פשוט שווה ל r.

[4] עבור מספר מרוכב כללי $r\cdot e^{\theta i}$ העלאה בריבע תגרום לסיבוב הנקודה בזווית θ ובנוסף להגדלת המרחק מהראשית פי r

[5] נסמן ב $\bar{z}$ את המספר ה"צמוד" לz. כלומר $z=a+bi\Rightarrow\bar{z}=a-bi$

הגדרה: פולינום מרוכב

עבור θ- פרמטר קבוע כלשהו. נסמן

$$x = e^{\theta i}$$

נקבל ש

$$x^k = e^{k\theta i}$$

נשים לב שעבור "סיבוב" אחד של x על מעגל היחידה x^k יבצע k "סיבובים". כלומר עבור הערכים $\theta \in [0, 2\pi]$ שעבורם המשתנה x יתן את כל הערכים על מעגל היחידה פעם אחת, x^k יתן את כל הערכים, כל אחד k פעמים.

העשרה

המשפט היסודי של האלגברה (עבור $\mathbb{C}$) לכל פולינום $p(x) \in \mathbb{R}[\mathrm{x}]$ (פולינום עם מקדמים מ $\mathbb{R}$) יש שורש. כלומר קיים x_0 כך ש $p(x_0) = 0$.

ההוכחה תושלם כשיהיה לי זמן (זה לא חלק מהחומר פשוט עדן אמר ש"חבל לדלג על זה. זאת הכוחה ממש יפה")

15.11.15 **טורי פורייה והתמרת פורייה על רגל אחת**

הערה: המבוא הזה, להבנתי, לא לגמרי נחוץ כדי להבין איך ולמה אלגוריתם FFT עובד. למי שאין כוח להתעמק בהקשר הרחב יותר של התמרת פורייה אפשר לדלג עד הכותרת "בחזרה לפולינומים"

נתעסק במרחב הוקטורי של פונקציות מהסוג

$$f : [-\pi, \pi] \to \mathbb{R}$$

כאשר חיבור פונקציות מוגדר

$$(f+g)(x) = f(x) + g(x)$$

וכפל בסקלר באופן דומה

$$(\lambda f)(x) = \lambda \cdot f(x)$$

הגדרה: מרחב מטרי הוא מרחב שבו הוספנו פונקציית מרחק בין איברים במרחב (מטרי מלשון "מטר" כלומר דרך למדוד מרחקים)

תזכורת: מרחב מכפלה פנימית הוא מרחב וקטורי שהוספנו לו אפשרות של כפל בעל תכונות מסויימות המכונה "מכפלה פנימית".

המכפלה הפנימית מאפשר גם למדוד אורכים/גדלים של איברים במרחב. גודל זה נקרא "נורמה".

ניתן להגדיר פונקציית מרחק/מטריקה בעזרת הנורמה - המרחק בין איברים יוגדר להיות ההפרש בין הנורמות.

במקרה שלנו: המכפלה הפנימית מוגדרת להיות[6]

$$\langle f,g\rangle=\int_{-\pi}^{\pi}f\left(x\right)g\left(x\right)dx$$

המטריקה שמקבלים ממכפלה פנימית זו נקראית מטריקה L_2 (עבור פונקציות) והיא מוגדרת באופן הבא

$$\left\|f-g\right\|_2=\sqrt{\int_{-\pi}^{\pi}\left|f\left(x\right)-g\left(x\right)\right|^2dx}$$

והיא למעשה מהווה הרחבה (עבור פונקציות) של פונקציית המרחק האוקלידית המוכרת בין שתי נקודות במרחב דו־מימדי.

משפט: אם נצמצם את המרחב לפונקציות "נחמדות"[7] אזי ניתן להגדיר למרחב בסיס אורתונורמלי[8] מהצורה

$$\frac{1}{2\sqrt{\pi}},\ \frac{1}{\sqrt{\pi}}\cos x,\ \frac{1}{\sqrt{\pi}}\sin x,\ \frac{1}{\sqrt{\pi}}\cos\left(2x\right),\ \frac{1}{\sqrt{\pi}}\sin\left(2x\right),...$$

או בכתיב אחר

$$B=\left\{\frac{1}{2\sqrt{\pi}}\right\}\cup\left\{\frac{1}{\sqrt{\pi}}\cos\left(nx\right)|n\in\mathbb{N}\right\}\cup\left\{\frac{1}{\sqrt{\pi}}\sin\left(nx\right)|n\in\mathbb{N}\right\}$$

המשמעות היא שכל פונקציה "נחמדה" ניתנן לייצג כסכום של $\frac{1}{2\sqrt{\pi}}$ ופונקציות מהצורה $\frac{1}{\sqrt{\pi}}\sin\left(nx\right)$, $\frac{1}{\sqrt{\pi}}\cos\left(nx\right)$ למעשה כפי שראינו באלגברה ניתן גם לבטא הצגה זו באופן מפורש

$$f\left(x\right)=\left(\int_{-\pi}^{\pi}\frac{1}{2\sqrt{\pi}}f\left(t\right)dt\right)\frac{1}{2\sqrt{\pi}}+$$

$$\sum_{\mathrm{n=1}}^{\infty}\left(\int_{-\pi}^{\pi}\frac{1}{\sqrt{\pi}}\cos\left(nt\right)f\left(t\right)dt\right)\frac{1}{\sqrt{\pi}}\cos\left(nx\right)+$$

$$\sum_{n=1}^{\infty}\left(\int_{-\pi}^{\pi}\frac{1}{\sqrt{\pi}}\sin\left(nt\right)f\left(t\right)dt\right)\frac{1}{\sqrt{\pi}}\sin\left(nx\right)$$

---

[6] ההגדרה היא רק עבור $[-\pi,\pi]$ כי צמצמנו את המרחב לתחום הזה בלבד

[7] חסומות, אינטגרביליות ובעלות מספר סופי של נקודות אי־רציפות וקיצון

[8] נשים לב שהמרחב, בניגוד לרוב המרחבים שהתעסקנו בהם באלגברה, הוא אינסוף מימדי, ולכן גם הבסיס יהיה אינסופי

טור פורייה

תזכורת: כאשר עסקנו בכפל פולינומים ראינו ש

$$a(x)\,b(x) = \sum c_k x^k$$

כאשר

$$c_k = \sum_{i=0}^{k} a_i b_{k-i}$$

ורצינו למצוא את המקדמים c_k

המטרה שלנו היא לנסות להמיר את הפולינומים שלנו לייצוג של טורי פורייה (נראה תכף מה זה) ובמקרה הזה יהיה לנו הרבה יותר קל לחשב את המקדמים.

ראינו ש $e^{xi} = cos(x) + i \cdot sin(x)$ על ידי העברת אגפים ועל פי זהויות טריגונומטריות נקבל ש

$$cos(x) = \frac{e^{xi} - e^{-xi}}{2}, sin(x) = \frac{e^{xi} - e^{-xi}}{2i}$$

נציב ערכים אלו בהצגה שראינו מקודם של $f(x)$ ונקבל

$$f(x) = \frac{1}{2\sqrt{\pi}} \sum_{n=-\infty}^{\infty} c_n e^{nxi}$$

כאשר

$$c_n = \int_{-\pi}^{\pi} e^{-nti} f(t)\, dt$$

הערה: הפונקציה שמקבלת f ומחזירה פונקציה $F(n) := c_n$ נקראת התמרת פורייה

כדי להרחב את ההגדרה ולייצג פונקציה מהצורה $f : \mathbb{R} \to \mathbb{R}$ מה שעושים זה להגדיר פונקציה מהצורה $f : [-k, k] \to \mathbb{R}$ ומשאיפים את k לאינסוף. לאחר חישובים רבים שנוותר עליהם מקבלים ש

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)\, e^{-2\pi x i \xi} d\xi$$

כאשר

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{2\pi x i \xi} dx$$

הגדרה: קונבולוציה

יהיו f,g פונקציות הקונבולוציה של f ו g מוגדרת להיות

$$(f\star g)(x)=\int_{-\infty}^{\infty}f(y)\,g(x-y)\,dy$$

אפשר להסתכל על קונבולוגציה כעל מעין הרחבה של כפל טורים למקרה הרציף.

משפט: משפט הקונבולוציה

$$(\hat{fg})(\xi)=\left(\hat{f}\star\hat{g}\right)(\xi)$$

$$(f\hat{\star}g)(\xi)=\left(\hat{f}\hat{g}\right)(\xi)$$

(ניסוח אחר אבל מעט שונה אומר שהתמרת פורייה של קונבולוציה של שתי פונקציות שווה למכפלת ההתמרות)

כפי שנראה בהמשך

בחזרה לפולינומים

למה: לכל $x_1,...,x_d$ שונים זה מזה ו $y_1,...,y_d$ (לאו דווקא שונים) קיים פולינום יחיד מדרגה $\geq d-1$ $p(x)$ כך ש

$$\forall 1\leq k\leq d:\ p(x_k)=y_k$$

מקרה פרטי של הלמה הזאת - בין כל שתי נקודות $(x_1,y_1),(x_2,y_2)$ עובר קיו ישר אחד (פולינום ממעהל 1)

מסקנה: ניתן לייצג פולינום ממעלה n על ידי רשימה של $x_1,...,x_{n+1}$ שונים זה מזה ורשימה של ערך הפולינום עבור ערכי x_i אלו.

נשים לב שעבור שני פולינומים $a(x),b(x)$ ותוצאת המכפלה שלהם $a(x)\,b(x)=c(x)$ אם נחשב בנקודה מסויימת את הערך של $y_a=a(x_0)$ ואת הערך של $y_b=b(x_0)$ הערך שנקבל מהפולינום $c(x)$ בנקודה זו שווה למכפלת התוצאות כלומר

$$c(x_0)=y_a y_b$$

ולכן אם נמיר את הייצוג הנוכחי של הפולינום כרשימת מקדמים לייצוג כרשימת ערכים (עבור רשימת x־ים שנבחרו מראש) נוכל לקבל את $c(x)$ בייצוג כרשימת ערכים בקלות רבה.

אם נבחר את רשימת הx־ים באופן אקראי, חישוב ערך פולינום ממעלה n בנקודה x_0 כלשהי דורש $\mathcal{O}(n)$ פעולות העלאה בחזקה ו $\mathcal{O}(n)$ פעולות חיבור. אפילו אם נתייחס אל העלאה בחזקה בתור פעולה בזמן $\mathcal{O}(1)$ עדיין נקבל שעלינו לבצע $\mathcal{O}(n)$ פעולות לכל x_i ויש לנו $n+1$ כאלה ולכן נקבל לכל הפחות $\mathcal{O}\left(n^2\right)$ כמו באלגוריתם הנאיבי.

הטריק הוא צריך למצוא רשימת x־ים שתאפשר לנו חישוב מהיר של ערכי $a(x),b(x)$ עבורם.

בהינתן פולינום נפצל אותו לחזקות זוגיות וחזקות אי־זוגיות

$$\begin{aligned} a(x) &= a_0 + a_1x + a_2x^2 + a_3x^3 + ... \\ &= \left(a_0 + a_2x^2 + a_4x^4 + ...\right) \\ &\quad + \left(a_1x + a_3x^3 + a_5x^5 + ...\right) \\ &= \left(a_0 + a_2x^2 + a_4x^4 + ...\right) \\ &\quad + x\left(a_1 + a_3x^2 + a_5x^4 + ...\right) \\ &= \left(a_0 + a_2\left(x^2\right) + a_4\left(x^2\right)^2 + ...\right) \\ &\quad + x\left(a_1 + a_3\left(x^2\right) + a_5\left(x^2\right)^2 + ...\right) \end{aligned}$$

כפי שניתן לראות קיבלנו בתור הסוגריים שני פולינומים שהמשתנה בהם הוא x^2 נקרא לראשון $a_{even}(x)$ ולשני $a_{odd}(x)$. כלומר

$$a_{odd}(x) = a_0 + a_2x + a_4x^2 + ...$$

$$a_{even}(x) = a_1 + a_3x + a_5x^2 + ...$$

הפולינום שלנו מתקבל מהם באופן הבא

$$a(x) = a_{even}\left(x^2\right) + x \cdot a_{odd}\left(x^2\right)$$

בכך פיצלנו את $a(x)$ לשני פולינומים צדרגה $\geq \frac{n}{2}$. השלב הבא והמתבקש יהיה להשתמש ברקורסיה כדי לחשב את הערכים עבור הפולינומים החדשים.

כעת נשים לב שמאחר והמשתנה של הפולינומים החדשים הוא x^2 הם נותנים את אותו ערך עבור $\pm x_0$ לכל x_0. משום כך אם נרכיב את רשימת ה־x־ים שלנו מזוגות של מספרים הופכיים, נצטרך לחשב רק חצי מהערכים ונחסוך זמן.

אילו x־ים נבחר?

נתבונן במקרים הבסיסיים:

אם $n = 1$ נבחר $x_1 = 1$

אם $n = 2$ אז נבחר $x_1 = 1, x_2 = -1$ וכפי שראינו נחסוך כך חצי מהפעולות על ידי רקורסיה.

אם $n = 4$ אנו מעוניינים בזוגות של ערכים שכל זוג הוא מסוג $\pm x$. בנוסף נרצה לאפשר קריאה לרקורסיה כלומר אם x ברשימה נרצה לחשב את הפולינומים $a_{odd}(x), a_{even}(x)$ עבור x^2. הפולינומים האלה הם מדרגה $n = 2$ ולכן נרצה שעבורם נקבל את הערכים ± 1 כלומר נרצה ש $x^2 = \pm 1$ ולכן נבחר

$$\begin{aligned} x^2 = & \pm 1 \\ \Rightarrow & \qquad x_1, x_2 = \sqrt{1} \\ & \qquad x_3, x_4 = \sqrt{-1} \\ \Rightarrow & \qquad x_1 = 1, x_2 = -1 \\ & \qquad x_3 = i, x_4 = -i \end{aligned}$$

כאמור בחרנו ב $n = 2$ את ± 1 כלומר x־ים המקיימים $x^2 = 1$ ולכן במקרה של $n = 4$ למעשה נבחר את 4 הערכים המקיימים

$$\left(x^2\right)^2 = x^4 = 1$$

או במילים אחרות נבחר את את השורשים מסדר 4 של 1 ב $\mathbb{C}$.

הסבר איטואיטיבי למה שהולך לקרות עכשיו - כמו שאנחנו רואים במקרים הפרטיים שהצגנו עכשיו, נרצה לפעול באופן רקורסיבי. ההפעלה של הפולינום על ערך x שקולה להפעלה של שני פולינומים (אחרים, מדרגה קטנה ממנו פי 2) על x^2. משום כך בכל כניסה פנימה ברקורסיה אנחנו מעלים בריבוע את הx־ים שלנו. מה שנרצה שיקרה הוא שבכל שלב נקבל ערכים שמתחלקים לזוגות מהצורה $\pm x$ כך שכאשר נעלה אותם בריבוע בשלב הבא התוצאה של הפעלת הפולינום על $(x)^2$ שווה לתוצאה של הפעלה שלו על $(-x)^2$ ונוכל לחשב רק אחד מהם ונחסוך חצי מהחישוב בכל רמה.

בסיס הרקורסיה כמובן נבחר לחשב פולינום ממעלה 1 על המספר 1. לכן בשלב אחד לפני נבחר שני ערכים שאם נעלה אותם בריבוע נקבל 1 כלומר את השורשים הריבועיים של 1 שהם ± 1. בשלב לפני כן נבחר את השורשים שלהם - $\pm 1, \pm i$ שהם למעשה השורשים של 1 מסדר 4 (כלומר $\sqrt[4]{1}$) ובשלב לפני כן את השורשים שלהם וכן הלאה. אם כך אם נתחיל מפולינום דרגה n, נניח ש n חזקה כלשהי של 2, נבחר בתור ערכים התחלתיים את $\vec{x} = \sqrt[n]{1}$ (נזכור שזו רשימה של n ערכים) וכאשר נקרא לרקורסיה ונעלה אותם בריבוע נקבל את $\vec{x^2} = \sqrt[\frac{n}{2}]{1}$ וכך נמשיך עד שנגיע לבסיס הרקורסיה 1 כפי שרצינו.

באופן פחות אינטואיטיבי ויותר פורמלי עבור n כלשהו נבחר את n השורישים מסדר n של 1. לשם כך נגדיר:

$$\omega = e^{\frac{1}{n}2\pi i}$$

כזכור הפרשנות הגיאומטרית של e^{xi} היא נקודה על מעגל היחידה בזווית x ולכן מאחר וזווית 2π מתארת מעגל שלם $x = {}^{2\pi}\!/_{n}$ מתארת זווית שהיא $\frac{1}{n}$ ממעגל שלם.

נשים לב שמתקיים

$$\omega^k = \left(e^{\frac{1}{n}2\pi i}\right)^k = e^{\frac{k}{n}2\pi i}$$

וקיבלנו, בדומה להסבר הקודם, זווית שהיא $\frac{n}{k}$ ממעגל שלם. ולכן אם נתבונן בקבוצה

$$\left\{\omega^k | k = 0, ..., n-1\right\}$$

נקבל בעצם חלוקה של מעגל היחידה לn חלקים כאשר הקבוצה היא אוסף הנקודות על המעגל המתאימות לחלוקה שכזאת.

לדוגמה - עבור $n = 4$ נקבל פשוט חלוקה של המעגל ל4 ולכן נקבל

$$\left\{e^{0i}, e^{\pi/2i}, e^{\pi i}, e^{3\pi/2i}\right\} = \{1, i, -1, -i\}$$

נבחר אם כן, בתור הרשימה $x_0, .., x_{n-1}$ את הרשימה $\omega^0, ..., \omega^{n-1}$.

עבור x־ים אלו מתקיים:

1. הערכים מתחלקים לזוגות כלומר לכל x^k קיים x^j יחיד שעבורו $-x^k = x^j$

$$x_{k+\frac{n}{2}} = \omega^{k+\frac{n}{2}} = \omega^k\omega^{\frac{n}{2}} = \omega^k e^{\frac{n}{2}\frac{1}{n}2\pi i} = \omega^k e^{\pi i} = -\omega^k = -x_k$$

2. גם אם נעלה בריבוע את הסדרה היא עדיין תתחלק לזוגות. כלומר לכל $\left(x^{k}\right)^{2}$ קיים איבר בסדרה x^{j} כך ש $\left(x^{k}\right)^{2}=-\left(x^{j}\right)^{2}$. נשים לב שכאשר אנו מעלים בריבוע את כל אברי הסדרה, בגלל תכונה 1 ומאחר ו $x=-y\Rightarrow x^{2}=y^{2}$ נקבל סדרה קצרה בחצי מזו שהיית לנו (בהנחה שאנחנו לא סופרים כפילויות).

19.11.15 בחזרה לכפל פולינומים

נתון פולינום

$$a\left(x\right)=a_{0}+a_{1}x+...+a_{n-1}x^{n-1}$$

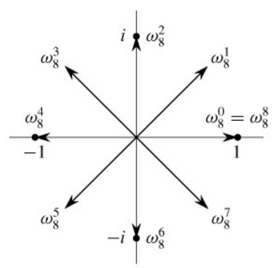
כזכור רצינו לבחור $x_{0},...,x_{n-1}$ ולחשב את

$$\forall0\leq i\leq n-1:\ A_{i}=a\left(x_{i}\right)$$

ובחרנו את

$$x_{k}=\omega^{k}=e^{\frac{2\pi i}{n}k}$$

כאשר ω הוא שורש יחידה מסדר n והסדרה למעשה יוצרת "חלוקה" של מעגל היחידה לn חלקים



איור 1: דוגמה עבור $n=$

נציב כעת ערך זה בפולינום ונקבל

$$A_{k}=a\left(\omega^{k}\right)=a\left(e^{\frac{2\pi i}{n}k}\right)=\sum_{j=0}^{n-1}a_{j}\left(e^{\frac{2\pi i}{n}k}\right)^{j}$$

[9]

---

[9]תזכורת (למי שראה) בטורי פורייה מקדמי פורייה היו דומים למדי לביטוי הזה

$$\hat{f}\left(k\right)=\int_{-\infty}^{\infty}f\left(x\right)e^{-2\pi kix}dx$$

ומכאן הקשר להתמרת פורייה. אנו עושים שימוש במשהו שדומה לגרסה דיסקרטית של התמרת פורייה

כפי שראינו ניתן להציג את הפולינום כצירוף של שני פולינומים במשתנה x^2

$$a\left(x\right)=a_{even}\left(x^2\right)+x\cdot a_{odd}\left(x^2\right)$$

עבור $x=\omega^k$ נקבל

$$a\left(\omega^k\right)=a_{even}\left(\left(\omega^k\right)^2\right)+\omega^k\cdot a_{odd}\left(\left(\omega^k\right)^2\right)$$

מאחר ו ω הוא שורש יחידה מסדר n ω^2 הוא שורש יחידה מסדר $\frac{n}{2}$ ולכן כפי שראינו מתקיים

$$\left(\omega^k\right)^2=\left(\omega^2\right)^k=\left(\omega^2\right)^{k+\frac{n}{2}}=\left(\omega^{k+\frac{n}{2}}\right)^2$$

ולכן כאשר נרצה לחשב את $a\left(\omega^k\right)$ נחשב רקורסיבית את $a_{even}\left(\left(\omega^k\right)^2\right),\ a_{odd}\left(\left(\omega^k\right)^2\right)$
כאשר בקריאה הרקורסיבית

$$a_{even}\left(\left(\omega^2\right)^k\right)=a_{even}\left(\left(\omega^2\right)^{k+\frac{n}{2}}\right)$$

$$a_{odd}\left(\left(\omega^2\right)^k\right)=a_{odd}\left(\left(\omega^2\right)^{k+\frac{n}{2}}\right)$$

מאחר ואנו מעוניינים לחשב את

$$a_{even}\left(\left(\omega^2\right)^k\right),a_{odd}\left(\left(\omega^2\right)^k\right),\ k=0,...,n-1$$

נוכל לחשבת

$$a_{even}\left(\left(\omega^2\right)^k\right),a_{odd}\left(\left(\omega^2\right)^k\right),\ k=0,...,\frac{n}{2}-1$$

ונקבל "בחינם", על פי השיוויון הקודם, את

$$a_{even}\left(\left(\omega^2\right)^k\right),a_{odd}\left(\left(\omega^2\right)^k\right),\ k=\frac{n}{2},...,n-1$$

ומכאן נקבל את

$$a\left(\omega^k\right)=a_{even}\left(\left(\omega^k\right)^2\right)+\omega^k\cdot a_{odd}\left(\left(\omega^k\right)^2\right),\ k=0,...,n-1$$

ובאופן הזה נחסוך חצי מהקריאות.

אלגוריתם $FFT\left(a\left(\cdot\right),\omega\right)$

כאשר $a\left(\cdot\right)$ נתון כסדרה של מקדמים. כאשר $\omega=e^{\frac{2\pi i}{n}k}$ כאשר n=מספר המקדמים.

- אם $\omega=1=e^{\frac{2\pi i}{1}k}$: כלומר $a\left(\cdot\right)=a_0$ ולכן נחזיר את a_0
- אחרת :

- נקרא ל $FFT\left(a_{even}\left(\cdot\right),\omega^{2}\right)$ נקבל חזרה את

$$a_{even}\left(\left(\omega^{2}\right)^{0}\right),a_{even}\left(\left(\omega^{2}\right)^{1}\right),...,a_{even}\left(\left(\omega^{2}\right)^{\frac{n}{2}-1}\right)$$

- נקרא ל$FFT\left(a_{odd}\left(\cdot\right),\omega^{2}\right)$ נקבל חזרה את

$$a_{odd}\left(\left(\omega^{2}\right)^{0}\right),a_{odd}\left(\left(\omega^{2}\right)^{1}\right),...,a_{odd}\left(\left(\omega^{2}\right)^{\frac{n}{2}-1}\right)$$

- כעת עבור $k=0,...,n-1$ נחשב ונחזיר:

$$a\left(\omega^{k}\right)\leftarrow a_{even}\left(\left(\omega^{2}\right)^{k}\right)+\omega^{k}\cdot a_{odd}\left(\left(\omega^{2}\right)^{k}\right)$$

כאשר את הערכים $k=\frac{n}{2},...,n-1$ מתקבלים ע"י

$$a_{even}\left(\left(\omega^{2}\right)^{k}\right)=a_{even}\left(\left(\omega^{2}\right)^{k+\frac{n}{2}}\right)$$

$$a_{odd}\left(\left(\omega^{2}\right)^{k}\right)=a_{odd}\left(\left(\omega^{2}\right)^{k+\frac{n}{2}}\right)$$

ניתוח: נסמן את מספר הקריאות הרקורסיביות עבור n ב $T\left(n\right)$. בכל שלב יש שתי קריאות רקורסיביות לחישוב $a_{even}\left(\omega^{2}\right),a_{odd}\left(\omega^{2}\right)$. כפי שראינו בכל אחת מהקריאות מספר הקריאות הנדרשות לצורך החישוב קטנות בחצי. כאשר אנו קוראים ל $a_{even}\left(\omega^{2}\right)$ אנו נדרשים לחשב רק עבור $k=0,...,\frac{n}{2}-1$.
מלבד זאת בכל שלב עלינו לחשב את הסכום $a_{even}\left(\left(\omega^{2}\right)^{k}\right)+\omega^{k}\cdot a_{odd}\left(\left(\omega^{2}\right)^{k}\right)$ עבור $k=0,...,n-1$ כלומר $\mathcal{O}\left(n\right)$ פעולות בכל שלב מעבר לקריאות הרקורסיביות. ולכן

$$T\left(n\right)=2T\left(\frac{n}{2}\right)+\mathcal{O}\left(n\right)\Rightarrow T\left(n\right)=\mathcal{O}\left(n\cdot log\left(n\right)\right)$$

דרך אחרת להבין את המעבר אחרון אפשר לראות שבכל שלב n קטן בחצי ולכן עומק הרקורסיה הוא $log\left(n\right)$. בכל רמה של עץ הרקורסיה יש 2 קריאות כל אחת מהן מגודל $\frac{n}{2}$ ולכן כל רמה לוקחת $\mathcal{O}\left(n\right)$ ולכן סה"כ כל הרמות יחד נקבל $\mathcal{O}\left(n\cdot log\left(n\right)\right)$

כעת, לאחר שחישבנו את ערכם של $a\left(\cdot\right)$ ו$b\left(\cdot\right)$ ב n נקודות שונים, נוכל לחשב את $c\left(\cdot\right)=a\left(\cdot\right)b\left(\cdot\right)$ באותם נקודות. כעת כל שנותר הוא לחשב את המקדמים של $c\left(\cdot\right)$ מתוך רשימה של n ערכים. כלומר, מה שנותר לעשות הוא אינטרפולציה.

נסמן ב FFT^{-1} את הפעולה ההפוכה ־מקבלים שערכוים (מה הפולינום, שאיננו ידוע, מחזיר עבור רשימה של x־ים) ומחזירים את מקדמי הפולינום. נראה בהמשך ש FFT^{-1} רצה באותו זמן ריצה של FFT.

נסכם את כל מה שראינו ונתאר את האלגוריתם השלם להכפלת פולינומים:

אלגוריתם כפל פולינומים

קלט: פולינומים $a(\cdot), b(\cdot)$ מדרגה $\geq d$

- נפעיל FFt על $a(\cdot), b(\cdot)$ ונקבל הערכה שלהם על n x־ים בזמן - $\mathcal{O}(n \cdot log(n))$ (נבהיר בהמשך מי זה n)
- נחשב $c(\omega^i) = a(\omega^i) b(\omega^i),\ i = 0, ..., n-1$ בזמן - $\mathcal{O}(n)$
- נפעיל FFT^{-1} על רשימת השערוכים של $c(\cdot)$ שקיבלנו ונקבל את מקדמי $c(\cdot)$ בזמן - $\mathcal{O}(n \cdot log(n))$

בסך הכל: זמן ריצה $\mathcal{O}(n \cdot log(n))$

הערה: $c(\cdot)$ מדרגה $\geq 2d$ ולכן (ממשפט האינטרפולציה שראינו) כדי לקבל אותו עלינו למצוא $2d+1$ שערוכים שלו. מצד שני, FFT עובד עם שערוך של מספר נקודות שהוא חזקה שלמה של 2. כלומר n־ מספר הנקודות (שאותו נכניס כקלט של FFT) צריך להיות 2^k המינימלי כך ש

$$2d + 1 \leq 2^k = n$$

במקרה הכי גרוע $2d+1$ גדול ב1 מחזקה שלמה של 2 (ולכן הפער עד החזקה הבאה הוא מקסימלי). במצב כזה $2d = 2^k$ עבור k כלשהו ולכן

$$2d + 1 = 2^k + 1 \Rightarrow n = 2^{k+1} = 2 \cdot 2^k = 2(2d) = 4d$$

כלומר גם במקרה גרוע

$$n = \mathcal{O}(d)$$

ומשום כך זמן הריצה שקיבלנו שקול לזמן ריצה

$$\mathcal{O}(d \cdot log(d))$$

FFT^{-1}

סימון: C_k (אות גדולה) מסמל ערך שמתקבל כאשר מציבים ω^k בפולינום $c(\cdot)$. c_k (אות קטנה) מסמל את המקדם הk בפולינום $c(\cdot)$.

כזכור קיבלנו $C_0, ..., C_{n-1}$ כאשר $C_k = c(\omega^k)$ ואנו רוצים למצוא את $c_0, ..., c_{n-1}$.
נכתוב את הפולינום באופן מפורש

$$c(\cdot) = \sum_{j=0}^{n-1} c_j x^j$$

והנתון שלנו משמעותו ש

$$c\left(\omega^0\right)=\sum_{j=0}^{n-1}c_j\left(\omega^0\right)^j=c_0+c_1+...+c_{n-1}=C_0$$

$$c\left(\omega^1\right)=\sum_{j=0}^{n-1}c_j\left(\omega^1\right)^j=c_0+c_1\omega+...+c_{n-1}\omega^{n-1}=C_1$$

$$c\left(\omega^2\right)=\sum_{j=0}^{n-1}c_j\left(\omega^2\right)^j=c_0+c_1\omega^2+...+c_{n-1}\left(\omega^2\right)^{n-1}=C_1$$

$$...$$

$$c\left(\omega^{n-1}\right)=\sum_{j=0}^{n-1}c_j\left(\omega^{n-1}\right)^j=c_0+c_1\omega^{n-1}+...+c_{n-1}\left(\omega^{n-1}\right)^{n-1}=C_{n-1}$$

קיבלנו n משוואות לינאריות ב n נעלמים $c_0,...,c_{n-1}$ ולכן נוכל לכתוב בכתיב אלגברי

$$\begin{pmatrix}C_0\\C_1\\.\\.\\.\\C_{n-1}\end{pmatrix}=\begin{pmatrix}1&1&1&.\;.\;.&1\\1&\omega&\omega^2&.\;.\;.&\omega^{n-1}\\1&\omega^2&\omega^4&.\;.\;.&\left(\omega^2\right)^{n-1}\\.&.&.&&.\\.&.&.&&.\\.&.&.&&.\\1&\omega^{n-1}&\left(\omega^{n-1}\right)^2&.\;.\;.&\left(\omega^{n-1}\right)^{n-1}\end{pmatrix}\cdot\begin{pmatrix}c_0\\c_1\\.\\.\\.\\c_{n-1}\end{pmatrix}$$

נשים לב שקיבלנו מטריצה $n\times n$ שמזכירה מאוד את מטריצת ואן־דרמונדה נסמנה ב W.

כדי לפתור את מערכת המשוואות ולקבל את $c_0,...,c_{n-1}$ כל שעלינו לעשות הוא לעפוך את המטריצה W שקיבלנו.

סתם כך להפוך מטריצה לוקח $\mathcal{O}\left(n^2\right)$ אלא שכאן מדובר במקרה מיוחד שבו אנו יודעים מה המטריצה ההופכית.

מתקיים

$$\begin{pmatrix}1&1&\cdots&1\\1&\omega&\cdots&\omega^{n-1}\\1&\omega^2&\cdots&\left(\omega^2\right)^{n-1}\\\vdots&\vdots&&\vdots\\1&\omega^{n-1}&\cdots&\left(\omega^{n-1}\right)^{n-1}\end{pmatrix}\cdot\begin{pmatrix}1&1&\cdots&1\\1&\omega^{-1}&\cdots&\omega^{-n+1}\\1&\omega^{-2}&\cdots&\left(\omega^{-2}\right)^{n-1}\\\vdots&\vdots&&\vdots\\1&\omega^{-n+1}&\cdots&\left(\omega^{-n+1}\right)^{n-1}\end{pmatrix}=\begin{pmatrix}n&0&\cdots&0\\0&n&&\\\vdots&&\ddots&\\0&\cdots&\cdots&n\end{pmatrix}$$

נימוק: האברים על האלכסון מתקבלים מכפל של שורה ועמודה בעלי אינקדס זהה

$$\sum_{j=0}^{n-1}\omega^{kj}\omega^{-kj}=\sum_{j=0}^{n-1}\omega^0=\sum_{j=0}^{n-1}1=n$$

שאר האיברים מתקבלים מכפל של עמודה k_1 ושורה k_2 כאשר $k_2 \neq k_1$ נסמן את התוצאה ב s

$$\sum_{j=0}^{n-1} \omega^{k_1 j} \omega^{-k_2 j} = \sum_{j=0}^{n-1} \omega^{j(k_1-k_2)} = s$$

נחשב כעת

$$\omega^{k_1-k_2} \cdot s = \sum_{j=0}^{n-1} \omega^{j(k_1-k_2)} \omega^{\omega k_1-k_2} = \sum_{j=0}^{n-1} \omega^{(j+1)(k_1-k_2)}$$

$$= [i = j+1] = \sum_{i=0}^{n} \omega^{i(k_1-k_2)}$$

ולבסוף נחשב

$$\left(\omega^{k_1-k_2} - 1\right) \cdot s = \omega^{k_1-k_2} \cdot s - \omega^{k_1-k_2} =$$

$$\sum_{i=0}^{n} \omega^{i(k_1-k_2)} - \sum_{j=0}^{n-1} \omega^{j(k_1-k_2)} = \omega^{n(k_1-k_2)} - \omega^{0(k_1-k_2)} = 0$$

המעבר האחרון נובע מקח $\omega^{0(k_1-k_2)} = 1$ ובנוסף מאחר ו ω הינו שורש יחידה מסדר n נקבל

$$\omega^{n(k_1-k_2)} = \left(\omega^n\right)^{(k_1-k_2)} = 1^{(k_1-k_2)} = 1$$

אבל הרי $k_2 \neq k_1$ ושניהם מספרים בין 0 ל $n-1$ ולכן $0 < k_1 - k_2 < n$ ומכאן נקבל שבהכרח

$$\omega^{k_1-k_2} - 1 \neq 0$$

אבל ראינו ש

$$\left(\omega^{k_1-k_2} - 1\right) \cdot s = 0$$

ולכן המסקנה היא ש

$$\sum_{j=0}^{n-1} \omega^{j(k_1-k_2)} = s = 0$$

בחזרה לאלגוריתם למציאת המקדמים:

כדי לקבל את המקדמים $c_0,...,c_{n-1}$ נחשב

$$\begin{pmatrix} c_0 \\ c_1 \\ . \\ . \\ . \\ c_{n-1} \end{pmatrix} \leftarrow \begin{pmatrix} 1 & 1 & 1 & . & . & . & 1 \\ 1 & \omega^{-1} & \omega^{-2} & . & . & . & \omega^{-n+1} \\ 1 & \omega^{-2} & \omega^{-4} & . & . & . & \left(\omega^{-2}\right)^{n-1} \\ . & . & . & & & & . \\ . & . & . & & & & . \\ . & . & . & & & & . \\ 1 & \omega^{-n+1} & \left(\omega^{-n+1}\right)^2 & . & . & . & \left(\omega^{-n+1}\right)^{n-1} \end{pmatrix} \cdot \begin{pmatrix} C_0 \\ C_1 \\ . \\ . \\ . \\ C_{n-1} \end{pmatrix}$$

כעת נשים לב שאם היינו רוצים לקבל את תוצאת המכפלה של המטריצה W בוקטור כלשהו המשמעות היא למעשה להפעיל FFT על הוקטור הזה (בתור וקטור של מקדמים של פולינום) וכך היינו מקבלים את תוצאת המכפלה באופן מהיר יותר. שהרי תוצאת FFT מקיימת את המשוואה האלגברית הראשונה שראינו.

באותה מידה נשים לב שנוכל לקבל את תוצאת ההכפלה של המטריצה W^{-1} בוקטור נתון ע"י הפעלה של FFT אלא שאת ω נחליף ב ω^{-1}.

ω^{-1} הוא גם שורש יחידה. אם נזכר במשמעות הגיאומטרית של ω - חילקנו את מעגל היחידה ל n חלקים ו ω הוא הנקודה הראשונה מעל הציר הממשי (זווית $\frac{2\pi}{n}$) והחזקות שלו היו שאר הנקודות (חזקה k נמצאת בזווית $\frac{2\pi}{n}k$). באותו אופן

$$\omega^{-1} = e^{-\frac{2\pi}{n}i} = e^{2\pi - \frac{2\pi}{n}} = e^{\frac{2\pi}{n}(n-1)} = \omega^{n-1}$$

במילים אחרות ω^{-1} הוא הנקודה שנקבל אם נזוז על מעגל היחידה בזווית $\frac{2\pi}{n}$ כלפי "מטה" (בכיוון ההפוך לזה הלכנו כדי לקבל את ω) והחזקות שלו יתנו את אותם ערכים שקיבלנו מהחזקות של ω בסדר הפוך. ולכן ניתן לבנות אלגוריתם FFT^{-1} באותו אופן בדיוק כמו FFT כאשר השינוי היחיד הוא שימוש ב ω^{-1} במקום ω (האלגוריתם זהה למעט החלפה זו).

מסקנה: השלב האחרון באלגוריתם הכפלת הפולינומים אכן דורש זמן $\mathcal{O}\left(n \cdot log\left(n\right)\right)$ כפי שרצינו.