אלגוריתמים 1 - 89-220
פתרון תרגיל 1*

1 בנובמבר 2020

שאלה 1 בתרגול הגדרנו את בעיית חישוב מרחקי האמינג בין טקסט לתבנית.
הציעו אלגוריתם הפותר את הבעיה כאשר הא"ב של הטקסט והתבנית הינו $\Sigma = \{0, 1, 2\}$.

תשובה: נסמן את $T = a_0a_1...a_{n-1}$ ואת $P = b_0b_1...b_{n-1}$ באותו אופן כפי שסימנו בתרגול. נאמר כי לכל $i \in \{0, ..., n-1\}$ ולכל $j \in \{0, ..., m-1\}$ האיברים a_i ו b_j הם המקדמים של החזקות x^i ו x^j, בהתאמה.
כפי שראינו בתרגול, אם נכפול את $T \cdot P^R$, נוכל לקבל את סכום המכפלה עבור כל היסט. לכן, נביט בטבלת הכפל שלנו:

$T \backslash P$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	4

כמובן, שהטבלה הנ"ל לא עוזרת לנו למצוא התאמות בצורה הנכונה. לכן, נתאים את הקלט באופן הבא:

1. פעם אחת, בכדי לסכום את כל ההתאמות בין האחדות, נחליף את כל תווי ה'2' ל'0' (נסמן את המחרוזות T_1, P_1), ונקבל את הטבלה הבאה:

$T_1 \backslash P_1$	0	1	$2 \to 0$
0	0	0	0
1	0	1	0
$2 \to 0$	0	0	0

ובעזרת האלגוריתם $FFT(T_1, P_1^R)$ נוכל לסכום התאמות של אחדות.

2. פעם נוספת, בכדי לסכום את כל ההתאמות בין האפסים, נחליף את כל תווי ה'2' וה'1' ל'0', ואת תווי ה'0' ל'1' (ולשם נוחות נסמן את המחרוזות (T_0, P_0), ונקבל את הטבלה הבאה:

$T_0 \backslash P_0$	$0 \to 1$	$1 \to 0$	$2 \to 0$
$0 \to 1$	1	0	0
$1 \to 0$	0	0	0
$2 \to 0$	0	0	0

*השאלון מנוסח בלשון זכר אך מכוון לסטודנטיות באותה המידה, עמכן הסליחה.

ובעזרת האלגוריתם $FFT(T_0, P_0^R)$ נוכל לסכום התאמות של אפסים.

3. לבסוף, בכדי לסכום את כל ההתאמות בין תווי ה'2', נחליף את כל תווי ה'2' ל'1', ואת תווי ה'1' ל'0', (ולשם נוחות נסמן את המחרוזות $(T_2, P_2$, ונקבל את הטבלה הבאה:

$T_2 \backslash P_2$	0	$1 \to 0$	$2 \to 1$
0	0	0	0
$1 \to 0$	0	0	0
$2 \to 1$	0	0	1

ובעזרת האלגוריתם $FFT(T_2, P_2^R)$ נוכל לסכום התאמות של '2'.

4. לכל היסט מסוים i, נסכום את כל ההתאמות של היסט זה בשם r, ונשמור עבורו את מרחק ההמינג, שהינו $m - r$. נחזיר את מערך הערכים שנוצר מהתא m עד התא ה n.

נכונות: כפי שראינו בתרגול, אם נסכום את כל ההתאמות עבור כל תו בנפרד יחדיו, נקבל את מרחק ההמינג המלא.

זמן ריצה: כיוון שאנו מפעילים 3 פעמים כפל פולינומים מהיר מדרגות שונות בעזרת FFT, ועוד עבודה ליניארית על מערכי הפלט של הפולינומים האלו (שאורכם $n + m$, למרות שאנו מתייחסים רק ל $n - m$ התאים האמצעיים כתוצאות של היסטים ־ הבינו מדוע), הרי שבסה"כ זמן הריצה חסום:

$$T_{Alg}(n, m) = \mathcal{O}(3 \cdot T(FFT(n, m) + n + m) = \mathcal{O}(3 \cdot n \cdot \log m + n + m) = \mathcal{O}(n \cdot \log m)$$

■

שאלה 2 (ע"פ מבחן תש"פ מועד ב'2)

בשאלה זו נעסוק במחרוזות מעל א"ב $\Sigma = \{1, 2, 3, \ldots, 20\}$.

נגדיר ששני תווים $a, b \in \Sigma$ הם "מתאימים־3" זה לזה אם ורק אם $|a - b| \leq 3$.

נכליל את ההגדרה ונאמר ששתי מחרוזות X, Y, שתיהן באורך n, הן "3־מתאימות" זו לזו אם ורק אם לכל $1 \leq i \leq n$ מתקיים $X[i]$ ו־$Y[i]$ מתאימים־3 זה לזה (כלומר, $|X[i] - Y[i]| \leq 3$).

<u>בעיית התאמת המחרוזות־3</u> מוגדרת באופן הבא:

<u>קלט</u>: מחרוזת טקסט T באורך n ומחרוזת תבנית P באורך $m \leq n$.

<u>פלט</u>: כל המקומות ב־T ש־P "3־מתאים" להם.

הצע אלגוריתם יעיל ככל הניתן אשר פותר את בעיית התאמת המחרוזות־3.

תשובה: תחילה נבחין כי $|\Sigma| = 20 \in \mathcal{O}(1)$. באופן דומה לשאלה הקודמת, נסמן $T = a_0 a_1 ... a_{n-1}$ ואת $P = b_0 b_1 ... b_{n-1}$, ונאמר כי לכל $i \in \{0, ..., n-1\}$ ולכל $j \in \{0, ..., m-1\}$ האיברים a_i ו b_j הם המקדמים של החזקות x^i ו x^j, בהתאמה. נבחין כי עבור כל איבר בקבוצה, $i \in \Sigma = \{1, ..., 20\}$, האיברים שהינם מתאימים־3 עבורו (כמובן בתנאי שהם קיימים, כלומר ־ לא חורגים מהטווח) הם האיברים:

$i - 3, i - 2, i - 1, i, i + 1, i + 2, i + 3$.

כלומר ־ היינו רוצים שעבור התו ה־i טבלת הכפל שלנו תחזיר '1' לכל תו מקבוצת התווים שתיארנו, ו'0' אחרת, אך לצערנו הדבר לא מתקיים בכפל רגיל בין הערכים האלו

(כצפוי). לכן, כדי לסכום 'התאמות־3' לכל תו, נרצה לבצע את ההתאמות הבאות: עבור כל תו σ שמופיע בתבנית P, בתוך התבנית נהפוך אותו ל'1' ואת שאר התווים (בדומה לפתרון בשאלה הקודמת), ובתוך הטקסט T את כל התווים שהם 'מתאימים־3' נהפוך ל'1', ואת שאר התווים ל'0' (ולמען הנוחות נסמנן בתור $(T_{\sigma_3}, P_{\sigma_3})$. באופן זה, נוכל לסכום 'התאמות־3' לכל תו בנפרד ולסכום על כולם. ננסח כאלגוריתם:

$:Alg(T,P)$

1. אתחל מערך של סכומי 'התאמות־3' בשם $result$, באורך $n+m-1$.
2. אתחל קב' ריקה $S \leftarrow \emptyset$.
3. לכל $\sigma \in \Sigma$ בצע:
   - הגדר את $T_{\sigma_3}, P_{\sigma_3}$ לפי ההסבר שבהקדמה.
   - בצע $FFT(T_{\sigma_3}, P^R_{\sigma_3})$ והוסף את ערכי התוצאה למערך $result$.
4. עבור $i=m$ עד n בצע:
   - אם $result[i]=m$:
     - $S \leftarrow S \cup \{i\}$.
5. החזר את הקבוצה S.

נכונות: כל אינדקס במערך $result$ שיש בו את הערך m, מראה לנו שישנה התאמה־3 מלאה עבור ההיסט הזה (ישנם m תווים וכל אחד מהם תרם 1 להתאמה), ולכן נוסיף אותו לקב' S. הערכים לכל היסט הם בהכרח נכונים כי בתבנית עצמה השארנו בכל פעם רק את מופעי התו שעבורו חיפשנו 'התאמות־3' (ושאר התבנית הפכו לאפסים).

זמן ריצה:

כיוון שיש כמות קבועה של תווים, ולכל תו נדרשת כמות ליניארית של עבודה על מערכי הקלט/פלט, ועוד הפעלה יחידה של FFT, נקבל שזמן הריצה הוא:

$$T_{Alg}(n,m) = \mathcal{O}\left(c\left(n+m\right) + 20 \cdot T\left(FFT\left(n,m\right)\right)\right) \in \mathcal{O}\left(n \log m\right) \quad \blacksquare$$

שאלה 3 נתון פולינום $A(x)$ מדרגה חסומה n. ניתן לחשב את הערך של $A(x_0)$ ע"י חלוקה של $A(x)$ בפולינום $(x-x_0)$ ע"מ לקבל פולינום $q(x)$ מדרגה חסומה $n-1$, ומספר r כך ש־$A(x)=q(x)(x-x_0)+r$. ברור כי $A(x_0)=r$. הראו כיצד בהינתן $A(x)$ ו־ x_0 , ניתן לחשב את $q(x)$ ואת r בזמן $O(n)$, כאשר $A(x)$ נתון ע"י המקדמים שלו.

תשובה: נסמן תחילה את $A(x)=\sum_{i=0}^{n-1} a_i x^i$, ואת $q(x)=\sum_{i=0}^{n-1} b_i x^i$.

ראינו בהרצאה כי כאשר פולינום מדרגה חסומה n מיוצג על ידי מקדמים, ניתן לחשב ערך של נקודה בזמן $\mathcal{O}(n)$ על ידי הצבה וחישוב כרגיל. לכן, בהינתן $A(x)$ ונק' מסוימת x_0, נוכל לחשב את $A(x_0)$, ולפי הנתון אנו יודעים כי התוצאה שווה ל r. כעת, משאנו יודעים מהי השארית r, נבחין כי מתקיים:

$$A(x)=q(x)(x-x_0)+r$$

$$\Rightarrow A(x)-r=q(x)(x-x_0)=x\cdot q(x)-x_0\cdot q(x)$$

כעת, נשים לב כי החזקה הקטנה ביותר של הביטוי $x \cdot q(x)$ הינה לפחות x^1, כלומר המקדם של האיבר החופשי x^0 בפולינום $A(x)-r$ הוא $x_0 \cdot b_0$, וכיוון ש x_0 ידוע ניתן לחלק

ולמצוא את b_0. באותו אופן, עבור המקדם של x, $x \cdot q(x)$ תורם בדיוק מקדם אחד (את b_0, כיוון שלפולינום x עצמו אין מקדם חופשי, והוא רק 'מזיז' את b_0 קדימה צעד אחד), ו $x_0 \cdot q(x)$ תורם את $x_0 b_1$ (כיוון שזהו כפל בקבוע, בלי הזזת המקדמים). למעשה, נפרוט את המקדמים של $A(x) - r$ לפי החזקות השונות:

$$a_0 - r = x_0 b_0$$
$$a_1 = b_0 - x_0 b_1$$
$$a_2 = b_1 - x_0 b_2$$

ובאופן כללי (מלבד האיבר הראשון):

$$a_i = b_{i-1} - x_0 b_i$$
$$\Rightarrow b_i = \frac{b_{i-1} - a_i}{x_0}$$

לכן, אם נתחיל לחשב מהאיבר '0', וכיוון שאנו יודעים מהו x_0, ונוכל בכל איטרציה i להשתמש ב b_{i-1} ובx_0 כדי לבודד את b_i מתוך המקדם a_i בפולינום $A(x) - r$. עבור כל ערך מדובר בפעולות קבועות, וכיוון שיש n משוואות נוכל למצוא את כל מקדמי $q(x)$ בזמן $\mathcal{O}(n)$. ■

שאלה 4 בהינתן פולינום $A(x)$ מדרגה חסומה n בייצוג ע"י n נקודות, ניתן לעשות אינטרפולציה ל־$A(x)$ לייצוג של מקדמים בעזרת נוסחת lagrange:

$$A(x) = \sum_{k=0}^{n-1} y_k \frac{\prod_{j \neq k}(x - x_j)}{\prod_{j \neq k}(x_k - x_j)}$$

הראו כיצד ניתן להשתמש בנוסחת lagrange על מנת לעשות אינטרפולציה ב־ $O(n^2)$ זמן.

רמז: קודם כל חשבו את $\prod(x - x_j)$ ואח"כ חלקו ב־ $(x - x_k)$ לפי הצורך. בשביל המונה של כל איבר בסכום - כדאי להשתמש בפתרון לשאלה 3.

תשובה: לאורך הפתרון נסמן $M_k(x) = \prod_{j \neq k}(x - x_j)$. נשים לב כי ההבדל בין $\prod_{j \neq k}(x - x_j)$ לבין $\prod_{j \neq k}(x_k - x_j)$ הוא האם הצבנו או לא את הערך x_k לתוך $M_k(x)$. לכן, נבצע הליך חישוב מקדים עבור תהליך האינטרפולציה כולה, ונחשב את $M(x) = \prod(x - x_j)$ עבור כל הנקודות. את ההליך הזה ניתן לבצע באופן איטרטיבי פשוט ־ ניקח את הפולינום שחושב עד עכשיו, נסמנו $m(x)$, ובכל פעם נכפול אותו ב $(x - x_j)$ עבור הנק' x_j הבאה. למעשה נחשב את הביטוי הבא: $m(x) \cdot (x - x_j) = xm(x) - x_j m(x)$

וזהו ביטוי שניתן לחישוב בזמן $\mathcal{O}\left(deg\left(m(x)\right)\right)$ בכל פעם. (כל אחד מהם ניתן לחישוב שכזה בנפרד ־ ראו שאתם מבינים מדוע)

לאחר מכן, עבור כל צמד (x_k, y_k), ניקח את $\prod(x - x_j)$ ונחלק אותו בפולינום $(x - x_k)$, בעזרת האלגוריתם משאלה 3, ונקבל את $M_k(x)$. כעת, כל שנשאר הוא לחשב ולקבל ערך מספרי עבור $M_k(x_k)$, לחלק את הפולינום $M_k(x)$ בתוצאה ולכפול ב y_k (או לחילופין, לכפול כל מקדם של $M_k(x)$ בערך $\frac{y_k}{M_k(x_k)}$). לבסוף, נסכום את ערכי התוצאה של נק' מסוימת לערכי כל הנקודות שכבר חושבו.

$Alg\left(\{(x_1, y_1), \ldots, (x_n, y_n)\}\right)$:

1. חשב את $M(x)$ כפי שתואר בהסבר.
2. אתחל מערך למקדמי התוצאה A.

3. לכל (x_k, y_k) בצע:
   - מצא את $M_k(x)$בעזרת חלוקת $\frac{M(x)}{(x-x_k)}$, בשימוש אלגוריתם משאלה 3.
   - חשב $z \leftarrow M_k(x_k)$.
   - חשב את $\frac{y_k}{z}$, וכפול את מקדמי $M_k(x)$ בערך זה.
   - הוסף את ערכי המקדמים שקיבלת למערך A באינדקסים המתאימים להם.
4. החזר את A.

נכונות:

מתקיימת עקב ההסבר המתמטי בתחילת הפתרון.

זמן הריצה:

שורה 1 מבצעת n איטרציות, ובכל פעם לוקח לה $\mathcal{O}\left(deg\left(m(x)\right)\right)$. נבחין כי $m(x)$ מתחיל את התהליך כפולינום בדרגה 1, ובכל איטרציה דרגתו עולה ב1, ולכן מדובר בזמן הריצה $\mathcal{O}(1+2+3\ldots+n)=\mathcal{O}(n^2)$.

עבור האתחול בשורה 2 נשלם $\mathcal{O}(n)$ כגודל מערך הפלט.

כל שורה בבלוק 3 עולה $\mathcal{O}(n)$ בנפרד ולכן כולן ביחד גם כן, והלולאה עצמה רצה n פעמים העלות תהיה $\mathcal{O}(n^2)$.

שורה 4, בדומה לשורה 2, תעלה לנו $\mathcal{O}(n)$.

בסה"כ ההליך כולו הינו $\mathcal{O}(n^2)$, כנדרש.

■

שאלה 5 תהי S מחרוזת באורך n מעל הא"ב $\{0,1\}$. שלשה של אינדקסים שונים (i,j,k) כך ש $1\leq i<j<k\leq n$ תקרא שלשה מיוחדת אם מתקיים $S[i]=S[j]=S[k]=1$ וגם $k-j=j-i$.

כתבו אלגוריתם יעיל המחזיר את **כמות** השלשות המיוחדות במחרוזת הנתונה S.

תשובה: נסמן את המחרוזת הבינארית שלנו $S=a_1a_2\ldots a_n$, כך שנתייחס לביט ה a_i שבה בתור המקדם של החזקה x^i. נרצה להביט בדוגמה כדי לקבל יותר אינטואיציה: נביט במחרוזת $S=10111$. השלשות שקיימות הן השלשות $\{1,3,5\},\{3,4,5\}$, ונסמן את $g(x)=x^5+x^4+x^3+x$ כפולינום מייצג למחרוזת. אם נכפול את $g(x)$ בעצמו, נקבל: $x^{10}+2x^9+3x^8+2x^7+3x^6+2x^4+x^2$.

נשים לב לתכונה מעניינת: בגלל שכל המקדמים בינאריים, ואנו מביטים רק על מקדמים שאינם 0, כאשר המקדם של x^i מוכפל בעצמו ־ הוא מוסיף אחד לסכום המקדם של החזקה x^{2i}, וכאשר שני מקדמים x^i וx^j מוכפלים זה בזה, הם מוסיפים 2 לסכום המקדם של החזקה x^{i+j}.

נבחין, כי שלשה מיוחד $1\leq i<j<k\leq n$ היא שלשת אחדות שמקיימת $k-j=j-i$, אך גם מקיימת $k+i=2j$. כלומר ־ אם נוכל למצוא את כל השלשות בהן שתי התכונות שתיארנו מקודם יתקיימו באותו אינדקס, נוכל לדעת על שלשות שנמצאות בו. חשוב לשים לב ולא לטעות בכך שיכולות להיות יותר משלשה אחת באותו אינדקס (נגיד, השלשות

$\{i,j,k\}$ והשלשה $\{i+1,j,k-1\}$, ששתיהן נסכמות באינדקס $2j$), אך עדיין כולן יורכבו מאותו אינדקס אמצעי שכפלנו אותו בעצמו, ועוד זוג הקצוות שתרמו 2 לסכום. לכן, עבור כל אינדקס בו הערך הינו אי־זוגי גדול ממש מ1, נחסיר מהמקדם אחד (עבור האינדקס האמצעי), ונחלק ב2 (למציאת מס' זוגות הקצה של כל שלשה). נכתוב כאלגוריתם:
:$Alg(S=a_1a_2...a_n)$

1. ייצג את המחרוזת כמקדמי פולינום.
2. בצע $FFT(S,S)$, והכנס את מקדמי המכפלה למערך B.
3. אתחל מונה $conuter \leftarrow 0$.
4. לכל i מ 1 עד $2n$:
   - אם $B[i]$ הינו ערך אי־זוגי וגדול מ־1:
     - $counter \leftarrow counter + \frac{B[i]-1}{2}$.
5. החזר את $counter$.

נכונות: נובעת מההסבר שניתן לפני האלגוריתם.
זמן ריצה:
ביצוע פעולת FFT יחידה, ובנוסף סריקה ליניארית עם פעולות קבועות בתוכה, ולכן חישוב הזמן הינו $T_{Alg}(n)=\mathcal{O}(n\cdot\log n+n)\in\mathcal{O}(n\cdot\log n)$. ■

שאלה 6 שורש של פולינום A הוא מספר r כך ש $A(r)=0$.
הצע אלגוריתם, שבהינתן n שורשים $(r_1,r_2,\ldots,r_n)$ של הפולינום A שהוא פולינום מדרגה n מוצא את מקדמי הפולינום A בזמן $O(n\log^2 n)$.

תשובה: תחילה נוכיח עבור מקרה בסיס:
כאשר $n=1$, השורש הינו r_1 בלבד, והפולינום היחיד מדרגה חסומה 2 שמקיים $p_1(r_1)=0$ הוא הפולינום $p_1(x)=x-r_1$.
כעת, האלגוריתם יפעל באופן הבא:
:$Alg(r_1,r_2,\ldots,r_n)$

1. אם $n=1$ החזר את הפולינום $p(x)=x-r_1$.
2. אחרת:
   - הפעל ברקורסיה ושמור $p_1(x)\leftarrow Alg(r_1,\ldots,r_{\lceil\frac{n}{2}\rceil})$.
   - הפעל ברקורסיה ושמור $p_2(x)\leftarrow Alg(r_{\lceil\frac{n}{2}\rceil+1},\ldots,r_n)$.
   - כפול את $p_1(x)$ עם $p_2(x)$ בעזרת FFT, והחזר תשובתו

נכונות: כיוון שיש לנו n שורשים, הם מגדירים לנו פולינום ייחודי מדרגה n. לכן, מספיק שניקח פולינומים שמוגדרים על ידי אותם השורשים, ונכפול אותם אחד בשני, בזה אחר זה, עד לקבלת ביטוי אחד, שהוא בהכרח הפולינום שאנחנו מחפשים. כיוון שהכפל אסוציאטיבי, ניתן לכפול בכל סדר שנבחר.

זמן הריצה: נבחין כי נוסחת הנסיגה שלנו מקיימת:
$T_{Alg}(n) = 2 \cdot T\left(\frac{n}{2}\right) + \frac{n}{2} \cdot \log \frac{n}{2}$.
הוכחת החסם עצמו מושארת כתרגיל. ■

בהצלחה!