שרה שירי לנדאו
305362121

ממן 13

שאלה 1

הפולינום: $p(x) = x^3 + 2x^2 - 3x - 1$, **וקטור המקדמים:** $(-1, -3, 2, 1)$

נשתמש ב - $p(x) = p_e(x^2) + xp_o(x^2)$

א. הרצת $FFT(\cdot, \omega_4)$ וקטור המקדמים:

$FFT\big((-1,-3,2,1), i\big)$
$\quad FFT\big((-1,2), -1\big)$
$\quad\quad FFT\big((-1), 1\big)\ return\ -\mathbf{1}$
$\quad\quad FFT\big((\ 2), 1\big)\ return\ \ \mathbf{2}$
$\quad return\ (p_e(1), p_e(-1)) = (-1+2, -1-2) = (\mathbf{1}, -\mathbf{3})$
$\quad FFT\big((-3,1), -1\big)$
$\quad\quad FFT\big((-3), 1\big)\ return\ -\mathbf{3}$
$\quad\quad FFT\big((\ 1), 1\big)\ return\ \ \mathbf{1}$
$\quad return\ (p_o(1), p_o(-1)) = (-3+1, -3-1) = (-\mathbf{2}, -\mathbf{4})$

$p(i^0) = p(\ 1) = p_e(1) + 1 * p_o(1) = 1 - 2 = -\mathbf{1}$
$p(i^1) = p(\ i) = p_e(i^2) + i * p_o(i^2) = p_e(-1) + i * p_o(-1) = -3 + i * (-4) = -\mathbf{3} - \mathbf{4i}$
$p(i^2) = p(-1) = p_e(1) - 1 * p_o(1) = 1 + 2 = \mathbf{3}$
$p(i^3) = p(-i) = p_e((-i)^2) - i * p_o((-i)^2) = p_e(-1) - i * p_o(-1) = -3 - i * (-4) = -\mathbf{3} + \mathbf{4i}$

Return (-1, -3-4i, 3, -3+4i)

ב. הרצת *INVERSE-FFT* ע"י הרצת - $FFT(\cdot, (\omega_4)^{-1})$ על הערכים שהתקבלו בסעיף א':

$FFT((-1, -3-4i, 3, -3+4i), -i)$
$\quad FFT\big((-1,3), -1\big)$
$\quad\quad FFT\big((-1), 1\big)\ return\ -\mathbf{1}$
$\quad\quad FFT\big((\ 3), 1\big)\ return\ \ \mathbf{3}$
$\quad return\ (p_e(1), p_e(-1)) = (-1+3, -1-3) = (\mathbf{2}, -\mathbf{4})$
$\quad FFT\big((-3-4i, -3+4i), -1\big)$
$\quad\quad FFT\big((-3-4i), 1\big)\ return\ -\mathbf{3} - \mathbf{4i}$
$\quad\quad FFT\big((-3+4i), 1\big)\ return\ -\mathbf{3} + \mathbf{4i}$
$\quad return\ (p_o(1), p_o(-1)) = (-3-4i-3+4i, -3-4i-(-3+4i)) = (-\mathbf{6}, -\mathbf{8i})$

$p((-i)^0) = p(\ 1) = p_e(1) + 1 * p_o(1) = 2 - 6 = -\mathbf{4}$
$p((-i)^1) = p(-i) = p_e((-i)^2) - i * p_o((-i)^2) = p_e(-1) - i * p_o(-1) = -4 - i * (-8i) = -\mathbf{12}$
$p((-i)^2) = p(-1) = p_e(1) - 1 * p_o(1) = 2 - (-6) = \mathbf{8}$
$p((-i)^3) = p(\ i) = p_e(i^2) + i * p_o(i^2) = p_e(-1) + i * p_o(-1) = -4 + i * (-8i) = \mathbf{4}$
Return (-4,-12,8,4) $\rightarrow$ $\mathbf{Divide\ by\ n = 4} \rightarrow (-\mathbf{1}, -\mathbf{3}, \mathbf{2}, \mathbf{1})$

לאחר הרצת *INVERSE-FFS* על הערכים שקיבלנו בסעיף א', קיבלנו את וקטור מקדמי הפולינום *p(x)* כנדרש.

שרה שירי לנדאו
305362121

יפה מאוד!
27/30

שאלה 2

תיאור האלגוריתם:

1. נחלק את הקלט ל $\frac{n}{k}$ בלוקים בגודל k סיביות כך ש: $B = \sum_{i=0}^{\frac{n}{k}-1} b_i \quad A = \sum_{i=0}^{\frac{n}{k}-1} a_i$. כאשר A,B מהווים ייצוג פולינומי של הקלט בבסיס 2, ו a_i, b_i מייצגים את הבלוק ה- i בגודל k סיביות בA,B בהתאמה.
2. נריץ FFT על A,B ונקבל את ערכיהם ב$\frac{2n}{k}$ נקודות שונות, נסמן את הווקטורים המתקבלים לאחר ריצת ה FFT ב- P(A) ו P(B) בהתאמה.
3. נכפול את הווקטורים שקיבלנו בשלב 2 כדי לקבל את הערכים של וקטור המכפלה - C. כך ש: $C = \mathrm{P}(A) * P(B)$
4. נריץ INVERSE-FFT על ווקטור המכפלה C שקבלנו בסעיף 3 כדי לקבל את מקדמי הפולינום c (נקבל את הערכים מוכפלים ב $\frac{2n}{k}$)
5. חישוב והחזרת הסכום ע"י החישוב: $csum = \sum_{i=0}^{\frac{2n}{k}-1} c_i * 2^{ik}$

נכונות האלגוריתם: האלגוריתם מתבסס על המרת המספרים לבסיס בינארי וכן על האלגוריתם להכפלת 2 פולינומים על ידי אלגוריתם FFT, לכן נכונות האלגוריתם נובעת מנכונות 2 האלגוריתמים הנ"ל.

ניתוח זמן ריצה:

1. חלוקת הקלט לבלוקים, תלויה לינארית במספר הסיביות ולכן $\boldsymbol{O(n)}$
2. הרצת FFT על 2 הווקטורים: גודל הקלט 2n, כל פעולת הכפלה היא θk^2, נציב $k = logn$ ונקבל:
$$T(2n) = 2(Tn) + \frac{n}{k} * \theta k^2 = 2(Tn) + \theta(nlogn) = \boldsymbol{\theta(nlog^2n)}$$
לא ברור על מה התבססת במעבר האחרון (-3)
3. הכפלת 2 וקטורים, יש $\frac{2n}{k}$ ערכים, וכל פעולת הכפלה היא k^2 לכן סה"כ $O\left(\frac{2n}{k} * k^2\right) = \boldsymbol{O(nk)}$
4. הרצת INVERSE-FFT בדומה לסעיף 2 $\boldsymbol{\theta(nlog^2n)}$
5. חישוב סכום: $O(\frac{n}{k})$

סה"כ זמן ריצה: $\boldsymbol{\theta(nlog^2n)}$ כנדרש.

אחלה
30/30

שאלה 3

תיאור האלגוריתם: בהתבסס על נוס' נגזרת לפולינום נגדיר 2 וקטורים: $A = (n!\,a_n, (n-1)!\,a_{n-1}, \ldots, 0!\,a_0)$, $B = (\frac{x_0^0}{0!}, \frac{x_0^1}{1!}, \ldots, \frac{x_0^n}{n!})$

כאשר A וקטור המקדמים של x_0 בכל נגזרות f, B וקטור חזקות של x_0 חלקי עצרת בגודל החזקה.

נבצע הכפלת וקטורים ע"י שימוש בכפל פולינומים בעזרת FFT (2 הרצות FFT, הרצת INVERSE-FFT) ונקבל וקטור המכיל את הערכים המבוקשים בסדר הפוך בn+1 איבריו הראשונים.

נכונות האלגוריתם: נובעת מנכונות האלגוריתם לכפל פולינומים בעזרת FFT

ניתוח זמן ריצה: יצירת וקטורים בזמן לינארי- $O(n)$, כפל פולינומים באמצעות FFT-$O(nlogn)$, סה"כ $\boldsymbol{O(nlogn)}$

שרה שירי לנדאו
305362121

שאלה 4

- מימוש ישיר של כפל מטריצות:
  - בכל שלב ברקורסיה: פירוק המטריצה ל4 תתי מטריצות של $\frac{n}{2} * \frac{n}{2}$
  - סה"כ פעולות כפל וחיבור: 8 פעולות כפל, 4 פעולות חיבור
  - סה"כ זמן ריצה לפי שיטת האב: $T(n) = 8T\left(\frac{n}{2}\right) + 4n^2 = \theta\left(n^{\log_2 8}\right) = \theta(n^3)$
- מימוש נתון של כפל מטריצות:
  - בכל שלב ברקורסיה: פירוק המטריצה ל4 תתי מטריצות של $\frac{n}{2} * \frac{n}{2}$
  - סה"כ פעולות כפל וחיבור: 7 פעולות כפל, 18 פעולות חיבור
  - סה"כ זמן ריצה לפי שיטת האב: $T(n) = 7T\left(\frac{n}{2}\right) + 18n^2 = \theta\left(n^{\log_2 7}\right) = \theta(n^{2.81})$

הוכחנו שבמימוש הנתון מספר הפעולות האלמנטריות של האלגוריתם הוא $\boldsymbol{\theta(n^{2.81})}$ בלבד, מש"ל.