<u>פתרון (מקוצר) לתרגיל מס' 1 באופטימיזציה גאומטרית</u>

*** הפתרונות המוצגים כאן נועדו לתת קווים מנחים לפתרון, אם כי הם בכל זאת מפורטים במידת מה. יש לציין כי במבחן או בהגשה של תרגיל, לעיתים יש לפרט/לנמק מעט יותר.**

1. בעיית ההכרעה – בהינתן מרחק x, האם המרחק ה-k-י הקטן ביותר בין זוגות נקודות ב-P ≤ ל-x.
עבור כל נקודה p ב-P, נמצא את הקטע c(p) באורך 2x שמרכזו ב-p. עבור כל נקודה p, נספור בכמה קטעים היא מוכלת – מייד נראה כיצד, (נוריד ממספר זה 1 עבור הקטע c(p) של הנקודה p עצמה), נסכם זאת על פני כל הנקודות ונחלק ב-2. זהו מס' זוגות הנקודות מ-P שמרחקם ≤ ל-x. נשווה מס' זה ל-k ונקבל תשובה לבעיית ההכרעה.
כדי לספור בכמה קטעים מוכלת כל נקודה, ראשית נמיין את נקודות P. זה ייתן גם את המיון עבור קצות הקטעים P+x, P-x. נמזג את שלושת הקבוצות P, P+x, P-x. ונקבל רשימה ממוינת אחת של נקודות P וקצות הקטעים. כעת נעבור על הרשימה משמאל לימין ונתחזק מונה. בכל פעם שניתקל בקצה קטע שמאלי נוסיף 1 למונה ובכל פעם שניתקל בקצה קטע ימני נוריד 1 מהמונה. כשניתקל בנקודה של P נוסיף את המונה הנוכחי לסכום הכולל. בסוף המעבר, כאמור, נוריד n מהסכום הכולל ונחלק ב-2.
כדי לפתור את בעיית האופטימיזציה ראשית נמיין את נקודות P (ו-P+x, P-x) בזמן O(n log n) ונשים לב שמיון זה אינו תלוי ב-x. מיון ראשוני זה נועד לכך שהרצת פרוצדורת ההכרעה בהמשך תיקח O(n) זמן.
כעת נסמלץ את אלגוריתם ההכרעה במרחק האופטימלי *x. לצורך כך נסמלץ מיון מקבילי של כל הנקודות וקצות הקטעים. לצורך הרצת האלגוריתם המקבילי נשתמש ב-O(n) מעבדים, כל מעבד ישתמש ב-O(log n) צעדים. בכל צעד נמיין את הערכים הקריטיים בזמן O(n log n) ונבצע ע"י חיפוש בינארי O(log n) קריאות לפרוצדורת ההכרעה שרצה בזמן O(n). סה"כ עלות: $O(n \log^2 n)$. ניתן לשפר ע"י שימוש בטכניקה של Cole לזמן O(n log n).
נשים לב שהמעבר על הנקודות הממויינות לאחר המיון אינו תלוי ב-*x ולכן אין צורך לסמלץ מעבר זה.
הערכים הקריטיים הינם מרחקים בין זוגות נקודות מ-P.

2. בעיית ההכרעה – בהינתן פרמטר r, האם קיימים שני ריבועים עם צלע באורך r שמכסים את P.
נשים לב כי הריבוע השמאלי מכיל על שפתו את הנקודה השמאלית ביותר מ-P, וכן את הנקודה העליונה או את הנקודה התחתונה ביותר מ-P. (אם הריבועים מופיעים **בדיוק** אחד מעל השני, ניתן להתייחס לכל אחד מהם כאל ריבוע שמאלי.) ננסה את שתי האפשרויות.
בה"כ נפתור את המקרה הראשון (בו הריבוע השמאלי מכיל את הנקודה השמאלית ביותר p_l ואת הנקודה העליונה ביותר p_t על שפתו).
נמצא את p_l ואת p_t ע"י פתירת בעיית min/max בזמן O(n), ונחשב (בזמן קבוע) את הריבוע c המכיל את p_l ו-p_t על שפתו.
עבור כל נקודה p ב-P נבדוק האם p נמצאת בתוך (או על שפת) c ונזכור את סט הנקודות P' שלא נמצאות ב-c. נמצא (בזמן לינארי) את ערכי max (בציר ה-x ובציר ה-y) של הנקודות ב-P', ונמצא את הריבוע המוגדר על ידם. נבדוק האם כל הנקודות ב-P' נמצאות בריבוע זה. סה"כ זמן ריצה: O(n).
האלגוריתם מבצע O(n) השוואות, בין ערכי x או y של נקודות ב-P (או ב-P') לבין r + ערכי x או y של נקודה אחרת ב-P. לכן הערכים הקריטיים של האלגוריתם הם מרחקי L_∞ בין זוגות נקודות מ-P.אחת הנקודות בכל זוג היא השמאלית או העליונה ולכן יש רק O(n) ערכים קריטיים. לכן ניתן למקבל את האלגוריתם עבור הערכים הקריטיים ע"י שימוש ב-O(n) מעבדים שכל אחד רץ בזמן O(1). (נשים לב כי מציאת ערכי מינימום ומקסימום אינה תלויה ב-r ולכן אין צורך למקבל זאת.)

נמיין את הערכים הקריטיים ב-$O(n \log n)$ זמן. נבצע באמצעות חיפוש בינארי $O(\log n)$ קריאות לפרוצדורת ההכרעה שרצה בזמן $O(n)$. ולכן סה"כ עלות האלגוריתם תהיה $O(n \log n)$.

3. (a) ההסתברות שקטע מסוים יכיל יותר מ- $\frac{cn}{r}\log r$ נקודות קלט, היא ההסתברות ש- $\frac{cn}{r}\log r+1$ נקודות לא ייבחרו, כלומר היא $(1-p)^{\frac{cn}{r}\log r+1}$. יש $O(n^2)$ אינטרולים פוטנציאלים. ההסתברות של אינטרול כזה להופיע במדגם היא לבחור את שני קצותיו ולא לבחור אף נקודה בתוכו, שהיא $\left(\frac{r}{n}\right)^2\left(1-\frac{r}{n}\right)^{\frac{cn}{r}\log r+1}$. לכן ההסתברות שאחד מ- $O(n^2)$ הקטעים הכבדים הפוטנציאלים ייבחר היא לכל היותר $O(n^2)\left(\frac{r}{n}\right)^2\left(1-\frac{r}{n}\right)^{\frac{cn}{r}\log r+1}$, ע"י פיתוח מתאים מקבלים שהסתברות זו $\le \frac{1}{r^{c-1}}$.

(b) נבחר דגימה אקראית בגודל $r=cn \log n$ מהקדקודים של המערך של L. מסעיף (a) נקבל שמס' הקדקודים בין כל זוג קדקודים עוקבים שבחרנו חסום בהסתברות גבוהה ע"י:

$$O(n)=\frac{c_1 n^2}{c_2 n\log n}\log(c_2 n\log n).$$

נבצע חיפוש בינארי על הקדקודים שבחרנו כדי למצוא את האינטרוול המכיל את הפתרון (באמצעות פרוצדורת ההכרעה שראינו בכיתה). (סה"כ זמן: $O(n \log^2 n)$). לאחר שמצאנו את האינטרוול המתאים נריץ line sweeping למציאת הפתרון באינטרוול זה, בהסתברות גבוהה בזמן של $O(n \log n)$. סה"כ תוחלת זמן ריצה: $O(n \log^2 n)$. (גם ביצוע הדגימה נכלל בחסם זה.)

4. בעיית ההכרעה – בהינתן רדיוס r, האם קיים מעגל ברדיוס r המכיל לפחות n-k מהנקודות. נוכל באופן שקול לבדוק האם קיימת נקודה המוכלת בלפחות (n-k) דיסקים שמרכזם נקודה ב-P ורדיוסם r.
לכל נקודה p ב-P, נבנה את הדיסק ברדיוס r שמרכזו ב-p, נסמנו $D_r(p)$. נשים לב כי מספיק לחפש נקודה המוכלת ב-n-k דיסקים על **שפות** הדיסקים, כיוון שעבור הרדיוס האופטימלי *r קיימת נקודה q ב-P שנמצאת על שפת הדיסק האופטימלי שרדיוסו *r, אחרת ניתן היה לכווץ את הדיסק האופטימלי (למעשה קיימות לפחות 3 נקודות על שפת הדיסק). ולכן מרכז הדיסק האופטימלי נמצא על שפת דיסק שמרכזו ב-q.
עבור כל מעגל $\partial D_r(p)$, נמצא את כל נקודות החיתוך של $\partial D_r(p)$ עם המעגלים האחרים. יש $O(n)$ נקודות חיתוך על כל מעגל. נמיין לכל מעגל את נקודות החיתוך שעליו עם כיוון השעון. כעת נטייל על כל מעגל עם כיוון השעון ונחזיק מונה הסופר את מס' הדיסקים המכילים את נק' החיתוך הנוכחית. נמצא את מס' הדיסקים המכילים את נקודת החיתוך הראשונה ב-brute force ב-$O(n)$. נמשיך לטייל על המעגל ועבור כל נק' חיתוך שנפגוש נחסיר 1 מהמונה או נוסיף 1 למונה, עד אשר נמצא נקודה שמוכלת ב-n-k דיסקים.
זמן ריצת פרוצדורת ההכרעה $O(n^2 \log n)$.
הערכים הקריטיים הם רדיוסי מעגלים ששלוש מנקודות הקלט נמצאות על שפתם.
כדי למקבל את פרוצדורת ההכרעה נשתמש ב-n^2 מעבדים C_{ij}, $1 \le i \le n$, $1 \le j \le n$.
המעבדים $C_{i1},\dots,C_{in}$ ישמשו לבדיקה אם יש נקודה בעומק n-k על הדיסק ה-i:
כל מעבד C_{ij} ייחשב את (לכל היותר 2) נקודות החיתוך של הדיסק ה-i עם הדיסק ה-j. בזמן מקבילי $O(1)$.

כל המעבדים $C_{i1},\ldots,C_{in}$ יבצעו מיון מקבילי של נקודות החיתוך על הדיסק ה-i, בזמן מקבילי O(log n).
ניתן להמשיך מכאן אולם נשים לב שכבר עד שלב זה עברנו על כל הערכים הקריטיים.
יש n^2 מעבדים שרצים ב-O(log n) צעדים, נמיין בכל צעד את הערכים הקריטיים בזמן $O(n^2 \log n)$ ונריץ בכל צעד את פרוצדורת ההכרעה על O(log n) ערכים קריטיים, ולכן נקבל אלגוריתם כולל עם זמן ריצה של $O(n^2 \log^3 n)$.
(כיוון שהאלגוריתם מבצע מיון ניתן להשתמש בשיפור של cole ולקבל אלגוריתם עם זמן ריצה $O(n^2 \log^2 n)$.)

5. פרוצדורת הכרעה: בונים ריבוע עם צלע 2δ סביב כל נקודה b של B כך ש-b במרכזו. בונים את איחוד הריבועים המתקבלים. סיבוכיות איחוד זה היא לינארית כיוון שכל צלע משתתפת לכל היותר פעם אחת באיחוד. מפרקים את האיחוד למלבנים זרים (וגם כאלו יהיו לכל היותר מס' לינארי). כעת יוצרים מערך M של n עותקים של האיחוד (המפורק) ע"י הזזת האיחוד ב-a- לכל a ב-A. כעת יש לבדוק האם חיתוך האיחודים אינו ריק. לצורך כך צריך לבדוק האם יש נקודה ב-M שהיא בעומק n. ניתן לבצע בדיקה זו ע"י שימוש ב- sweep או ב-segment tree. סיבוכיות M היא $O(n^2)$ ולכן סיבוכיות הפרוצדורה היא $O(n^2 \log n)$.