

Assignment 2 הסבר העבודה

המימוש שלנו הוא שתי רשימות ממוינות מקושרות כאשר לכל איבר ברשימה יש שדות:

Data המכיל Point

Next המצביע לאיבר הבא ברשימה

Prev המצביע לאיבר הקודם ברשימה

Other המצביע לcontinar המכיל את אותה הנקודה ברשימה הממוינת לפי הציר השני

בנאי: מאתחל שני רשימות $O(1)$

addPoint: מוסיף נקודה למקום המתאים בשתי הרשימות בעזרת השיטה Add של הרשימה כאשר היא רצה על הרשימה עד המקום המתאים ומוסיפה את האיבר החדש.

לכן הזמן ריצה של הוספה הוא $O(n)$

getPointsInRangeRegAxis: הפונקציה מתקדמת על הרשימה המתאימה (לפי axis) מההתחלה וגם מהסוף עד שמגיעה לטווח הרצוי, יוצרת מערך חדש ומכניסה את כל הנקודות בטווח לתוכו.

זמן הריצה הוא $O(n)$

getPointsInRangeOppAxis: הפונקציה עוברת על המערך הממוין בציר not axis ועבור כל חוליה בודקת האם קואורדינטת ה axis שלה בתחום הנדרש אם כן מוסיפה את הנקודה למערך

זמן ריצה של הפונקציה הוא $O(n)$

narrowRange: הפונקציה רצה מתחילת הרשימה ולאחר מכן מסופה עד שהיא מגיעה לטווח הרצוי כאשר בכל מעבר היא מוחקת את החוליה ברשימה ואת החוליה עם אותו point ברשימה הממוינת לפי הציר ההשני בעזרת הפונקציה unlink שזמן הריצה שלה הוא $O(1)$ ולכן הזמן הכולל של הפונקציה הוא $O(n)$

getLargesAxis: מכיוון שבמבני נתונים שלנו יש שדה first ו-last לכן בעזרת גישה ישירה נוכל לחשב את הטווח הנקודות בכל ציר ולהחזיר תשובה מי הוא הציר עם הטווח הגדול ביותר ב $O(1)$

getDensity: מכיוון שיש גישה ישירה לקצוות הרשימה ויש שדה size שמתעדכן כל הכנסה ומחיקה החישוב נעשה ב $O(1)$

getMedian: בעזרת הפונקציה size (שמתעדכנת כל הכנסה או מחיקה) מחשבים את מיקומו של ה-median עושים מעבר על הרשימה עד מגעים לאליו ומחזרים את החוליה זמן הריצה של הפונקציה הוא $\frac{n}{2}$ כלומר $O(n)$

```
Split(int value, Boolean axis)
Container split=null
Container f=byXlist.first
Container l=byXlist.last
While (f.data.getX>l.data.getX)
    if (f.data.getX>=value)
        split=f
        break
    if (l.data.getX<=value)
```

```

    split=l
    break
    f=f.getNext
    l=l.getPrev
    if (split ==null)
        return null
    new array={list from byXlist.first to split.prev ,list from split to byXlist.last}
    return array

```

*This code was written only for axis = true but needed to check in every line that has X in it about the possibility that axis=false.

nearsetPair: הפונקציה מממשת את האלגוריתם המוצע בחלק 4.2, כאשר חלק ממימוש זה כולל קריאה לפונקציה **nearestPairInStrip**.

nearestPairInStrip: מחשבת את הקואורדינטה המקסימלית והמינימלית לפי הציר המתבקש, על פי קואורדינטות אלה מוצאת את החוליות שבקצוות הטווח, ומחליטה לפי גודל הרצועה האם כדאי לה להמיר למערך את הרשימה או להמיר את הטווח שמצאה. היא עוברת על אחת מהרשימות (אם מדובר על תת הטווח, היא ממיינת אותו תחילה לפי הציר השני), עבור כל נק' היא מחשבת את המרחק המינימלי עם שבעת החוליות הבאות אחריה ושומרת את מרחק זה ואת זוג הנק' אשר הן במרחק מינימאלי אחת מהשנייה. לבסוף היא מחזירה זוג זה.

ניתוח זמן ריצה:

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

ניחוש: $T(n) = n \cdot \log(n)$

נוכיח בעזרת אינדוקציה:

בסיס: $n = 2$

$$T(n) = 2T(1) + 2 = 4$$

$$4 \leq c \cdot 2 \log 2 = 2c$$

מתקיים עבור $c \geq 2$

הנחת האינדוקציה: נניח שהטענה מתקיימת עבור $2 \leq k < n$ כלומר $T(k) = ck \cdot \log(k)$

צעד האינדוקציה: צריך להוכיח שהטענה מתקיימת עבור $k=n$ כלומר $T(n) = cn \cdot \log(n)$

$$T(n) = 2T\left(\frac{n}{2}\right) + n \leq 2c \cdot \left(\frac{n}{2}\right) \log\left(\frac{n}{2}\right) + n$$

$$2c \cdot \left(\frac{n}{2}\right) \log\left(\frac{n}{2}\right) + n = c \cdot n(\log n - \log 2) + n = c \cdot n \log(n) - cn + n$$

$$c \cdot n \log(n) + n(1 - c) \leq c \cdot n \log(n)$$

מתקיים כאשר $0 \leq 1 - c$ כלומר $c \geq 1$

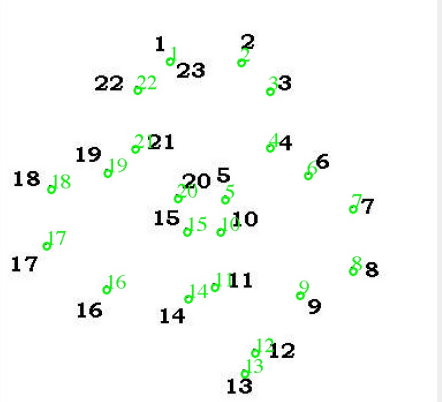
הנחנו בבסיס האינדוקציה שהביטוי מתקיים עבור $c \geq 2$ לכן זאת התשובה.

בונוס: השימוש שבחרנו למימוש הממשק הוא פותר ציורים של נק' (מתיחת קווים מנק' לנק' עד שיוצא ציור). בהתאמה קטנה אפשר להוסיף לממשק הגרפי את היכולת למתוח קווים ישרים.

Assignment no. 5

Input Menu Test

Test Your Point



Input list

- 1;147;58
- 2;209;59
- 3;234;84
- 4;234;133
- 5;195;178
- 6;267;157
- 7;306;186
- 8;306;240
- 9;260;261

Output list

- 1(147, 58)
- 2(209, 59)
- 22(119, 83)
- 3(234, 84)
- 4(234, 133)
- 21(117, 134)
- 19(93, 155)
- 6(267, 157)
- 18(144, 160)