# תרגול מספר 2

## טיטוא Plane Sweep

בכל אלגוריתם טיטוא אנחנו צריכים להציג את המרכיבים הבאים:

- האירועים, מבנה תור האירועים
- האינווריאנטה
- הסטטוס ומבנהו
- טיפול באירועים

פתרנו את השאלות הבאות:

1. (Point set maxima) נגדיר שנקודה $p=(p_1,p_2)$ שולטת על נקודה $q=(q_1,q_2)$ אם $q_1<p_1,\ \ q_2<p_2$ . נגדיר שנקודה $p\in P$ היא מקסימלית ב- P אם לא קיימת נקודה $q\in P$ ששולטת עליה. בהינתן קבוצה P של $n$ נקודות, מצא את כל הנקודות המקסימליות בזמן $O(n\log n)$.
2. נתונים $n$ סגמנטים זרים ו- $m$ מעגלים זרים במישור (המעגלים יכולים להכיל זה את זה, אך אינם נחתכים). מצא באופן יעיל את כל המעגלים שאינם נחתכים עם אף סגמנט.

### **פיתרון לשאלה 1**

נבצע אלגוריתם טיטוא.

האירוע: אירוע נקודה.

תור האירועים: נשמור תור אירועים ברשימה. תור האירועים יאותחל עם כל הנקודות, ממוינות לפי שיעור y יורד (אם אין מצב כללי- במקרה של נקודות עם אותו ערך y, נכניס שיעורי x קטנים יותר לתור קודם).

האינווריאנטה: בכל שלב בריצת האלגוריתם, זוהו כל הנקודות המקסימליות שנמצאות מעל ישר הטיטוא.

הסטטוס: בכל שלב בריצת האלגוריתם, נשמור במשתנה max את שיעור ה- x המקסימלי של נקודה שהשתתפה באירוע.

טיפול באירוע נקודה: אם שיעור ה- x של הנקודה גדול מ- max, נוציא את הנקודה לפלט ונעדכן את max להיות שיעור ה- x של הנקודה.

ניתוח זמן ריצה: המיון לקראת ההכנסה לתור לוקח $O(n\log n)$. יש $n$ אירועים והטיפול בכל אירוע לוקח $O(1)$. אז בסה"כ זמן הריצה הוא $O(n\log n)$.

הוכחת נכונות: נקודה היא מקסימלית $\Leftrightarrow$ אין נקודה שנמצאת מימינה ומעליה $\Leftrightarrow$ כל הנקודות מעליה נמצאות משמאלה או ישירות מעליה. לכן, בזמן טיטוא של המישור מלמעלה למטה, נקודה חדשה $p$ שאנחנו מגיעים אליה היא מקסימלית $\Leftrightarrow$ היא הייתה הכי ימנית עד כה.

אז בשביל לבדוק אם נקודה חדשה היא מקסימלית, אכן מספיק לבדוק את שיעור ה- x שלה לעומת שיעור ה- x המקסימלי של הנקודות הקודמות. לכן אלגוריתם הטיטוא משיג את המבוקש.

## **<u>פיתרון לשאלה 2</u>**

נבצע אלגוריתם טיטוא.

<u>האירועים:</u>

- אירוע תחילת סגמנט
- אירוע סוף סגמנט
- אירוע תחילת מעגל
- אירוע סוף מעגל

<u>תור האירועים:</u> רשימה. תור האירועים יאותחל עם כל אירועי תחילת וסוף הסגמנטים והמעגלים, לאחר מיון לקסיקוגרפי ראשוני שלהם עם תחילת האלגוריתם.

<u>האינווריאנטה:</u> בכל שלב בריצת האלגוריתם:

- זוהו המעגלים (חצאי המעגלים) שהתחילו מעל ישר הטיטוא ונחתכו עם סגמנט מעל ישר הטיטוא.
- עבור כל מעגל שהסתיים מעל ישר הטיטוא, זוהה אם הוא נחתך עם סגמנט או לא.

<u>הסטטוס:</u> בכל שלב בריצת האלגוריתם, נשמור את חצאי המעגלים ואת הסגמנטים אותם ישר הטיטוא חותך, לפי הסדר שלהם משמאל לימין. זאת אלא אם אנו כבר יודעים על חצי מעגל שהוא נחתך עם סגמנט, ואז נוציא אותו מהסטטוס בהקדם.

<u>מבנה הסטטוס:</u> נשמור את הסדר של חצאי המעגלים והסגמנטים בעץ חיפוש בינארי מאוזן.

<u>טיפול באירועים:</u>

- אירוע תחילת סגמנט:
  נוסיף את הסגמנט לעץ הסטטוס במקום המתאים. נוצרו שכנויות חדשות של הסגמנט עם אחרים. נבדוק שכנויות אלו: אם נוצרה שכנות חדשה של סגמנט-חצי מעגל והם נחתכים בהמשך, נוציא את חצי המעגל מהסטטוס. לאחר הסרת חצי המעגל מהסטטוס יתכן שתיווצר שכנות חדשה של סגמנט - חצי מעגל. במצב כזה נחזור על התהליך עד שלא נצטרך להוציא חצי מעגל מהסטטוס כתוצאה מהשכנות שנוצרה.
- אירוע סוף סגמנט:
  נוריד את הסגמנט מעץ הסטטוס. נוצרה שכנות חדשה, נבדוק אותה כמקודם.
- אירוע תחילת מעגל:
  נוסיף שני חצאי מעגל- חצי ימני וחצי שמאלי, לעץ הסטטוס במקום המתאים. נוצרו שכנויות חדשות- נבדוק אותן כמקודם.
- אירוע סוף מעגל:
  אם שני חצאי המעגל עדיין נמצאים בסטטוס, נדווח על המעגל כמעגל שלא נחתך עם אף סגמנט. בכל מקרה נוריד את כל חצאי המעגל שנותרו בסטטוס. נוצרה שכנות חדשה- נבדוק אותה.

<u>ניתוח זמן ריצה:</u>

גודל תור האירועים המקסימלי- $O(m+n)$, זהו גם מס' האירועים.

נשים לב שלכל היותר $2m$ פעמים אנחנו מוצאים חיתוך בין סגמנט לחצי מעגל- פעם אחת לכל חצי מעגל (ואז חצי המעגל ימחר מהסטטוס ולא יחזור אליו יותר). לכן בסה"כ הוצאות חצאי מעגלים לוקחות זמן $O\left(m\log\left(m+n\right)\right)$. שאר הבדיקות של השכנויות החדשות שכל הזמן נוצרות הן ב- $O\left(1\right)$.

פרט למקרה זה, כל אירוע מבצע מספר סופי של הכנסות והוצאות מעץ הסטטוס. לכן כל אחד מהם רץ בזמן $O\left(\log\left(m+n\right)\right)$, ובסה"כ זמן הריצה של האלגוריתם כולו הוא $O\left(\left(m+n\right)\log\left(m+n\right)\right)$.