-לכל קונפיגורציה משקל w = מס' האובייקטים שנמצאים איתה בקונפליקט (מס' הנק' שרואות את הסגמנט).

-מדגם אקראי R ע"י בחירת כל האובייקטים בהסתברות $p=\frac{1}{w}$.

-הסתכלו על מס' הקונפיגורציות שמופיעות ב-R במשקל 0. $E(N_0(R))$

$$E(N_0(R))=\sum_{k} N_k p^d(1-p)^k \ge p^d\sum_{k=0}^{w}(1-p)^k N_k \ge p^d(1-p)^w N_{\le w}$$

$$N_{\le w}=O(w^d\cdot E(N_0(R)))=O(w^d\cdot N_0(\tfrac{n}{w}))$$

1.12.08

*דוגמא לשימוש:

n ישרים במישור. ⇐ לא אנכיים, במצב כללי.

צ"ל: חסום את מס' נק' החיתוך ביניהם, שמתחתיהן $\ge w$ ישרים

אובייקטים = ישרים

קונפיגורציה = נק' חיתוך

$d=2$

קונפליקט = הישר מתחת לנק'

$N_0(m)\le m-1\le m$

$$\Longrightarrow N_{\le w}(n)=O(w^2\tfrac{n}{w})=O(nw)$$

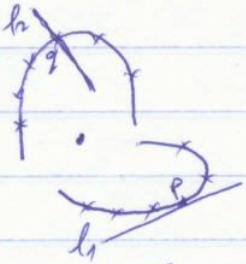
* קמור במימדים שבהם $d \ge 4$:

- סיבוכיות מספר הפאות ה-$(d-1)$ מימדיות (עבור d מימד) היא: $\Theta(n^{\lfloor d/2 \rfloor})$.

$\Rightarrow d=2,3 \to \Theta(n)$, $d=4,5 \to \Theta(n^2)$

דוגמא:

נבנה קמור ב-4 מימדים עם סיבוכיות ריבועית.
ניקח שני מישורים מאונכים זה לזה דרך הראשית.
קואורדי': x,y,z,w. $\Pi_1: z=w=0$, $\Pi_2: x=y=0$.
על כל מישור נפזר $\frac{n}{2}$ נק' על חצי מעגל ברדיוס 1 סביב הראשית.
לכל נק' p על המעגל הראשון ו-q במעגל השני - pq צלע של הקמור.
$\frac{n^2}{4}$ צלעות.



l_1 תומך לקמור, עובר דרך p
l_2 תומך לקמור, עובר דרך q
קיים על-מישור יחיד העובר דרך l_1, l_2
נראה על-מישור שנוגע ב-p, q ושאר הנק' בצד אחד שלו.

$$\left.\begin{array}{l} l_1: Ax+By+1 \ge 0 \\ l_2: Dz+Ew+1 \ge 0 \end{array}\right\} \Longrightarrow Ax+By+Cz+Dw+1 \ge 0$$

* אפשר להפעיל את האלג' הרנדומי האינקרמנטלי לכל מימד, ותוחלת זמן הריצה שלו תהיה $O(n^{\lfloor d/2 \rfloor})$, $d \ge 4$.

טאטוא ע"י ישר - Line Sweeping

טכניקה כללית לפתרון בעיות במישור.

נסביר ע"י דוגמא:

1. נתון אוסף קטעים, S, ישרים במישור.
   לבדוק האם קיים חיתוך ביניהם.
2. למצוא את כל החיתוכים בין הקטעים הנ"ל. במקרה הגרוע $\Theta(n^2)$.

עבור בעיה ②, הזמן צריך להיות תלוי ב-n וב-k = מס' החיתוכים.

לפתרון הבעיה ניעזר בטאטוא (קו ישר אנכי), ונזיז אותו
ימינה על פני המישור עבור $-\infty < x < \infty$.

בתנועתו ימינה, L (הישר) נתקל בנק' קריטיות:
קטע חדש מופיע - נק' הקצה השמאלית
קטע קיים נעלם - " " הימנית

בין כל שני אירועים קריטיים עוקבים, L יחזיק מידע לגבי מה
שהוא חותך ברשימת הקטעים שהוא חותך בסדר עולה.

* **רשימת x, X-structure:**

רשימת האירועים הקריטיים לפי סדר עולה, בדוגמא זאת
רשימה סטטית שאפשר להכין אותה מראש (ע"י מיון), אבל
בד"כ היא יכולה להיות דינאמית.

* **רשימת y, Y-structure:**

רשימת הקטעים ש-L חותך לפי סדר עולה של y.
רשימה דינמית עם פעולות עדכון: insert, delete, swap - החלפת
סדר של שני קטעים סמוכים.
הרשימה ממומשת ע"י עץ חיפוש מאוזן, כל עדכון מתבצע ב-$O(\log n)$.

* **האלגוריתם:**

1. ממיין את נק' הקצה של הקטעים לתוך רשימת x, $O(n\log n)$.
2. מאתחל את רשימת y לרשימה ריקה (עץ חיפוש ריק). ← ריק כי [illegible] חסומים ב-x
3. עובר על רשימת x לפי הסדר, ובכל אירוע מוסיף את הקטע
   המתאים (עבור נק' קצה שמאלית) או מוחק אותו (עבור נק' קצה

ימנית). $2n$ עדכונים, כ״א עולה $O(\log n)$.
⇐ סיבוכיות כוללת: $O(n\log n)$.
בכל רגע, אם האירוע הבא הוא חיתוך, זה חייב לקרות בין
זוג קטעים סמוכים (ברשימת y).
לכן, כל פעם שיש שינוי (דיסקרטי) ברשימת y ניתן לבדוק את
כל הזוגות העוקבים. ← מ״דית נותן $\Theta(n^2)$.
למעשה לאחר כל שינוי דיסקרטי מספיק לבדוק רק את הזוגות
העוקבים החדשים ברשימה.
עבור הוספה יש לכל היותר 2 זוגות חדשים, ועבור מחיקה - אחד.
לכן, אחרי כל שינוי נבצע $O(1)$ בדיקות חיתוך ונעצור כשנגלה כזה.

כאן צריך לחשב את נק׳ החיתוך כדי להכניס לרשימת x

* בדיקת חיתוך: abc, abd - [illegible]
4 בדיקות פנייה שמאלה. cda, cdb - " "
* כדי למצוא את כל k נק׳ החיתוך, נריץ את אותו האלג׳,
וכל פעם שיתגלה חיתוך - נקבל אירוע חדש (נק׳ החיתוך) ונכניס
אותו לרשימת x במקום המתאים (אם כבר הוא לא עבר כבר).
כעת, רשימת x השתנה לדינמית.
⇐ רשימת x הופכת לתור עדיפויות עם deletemin, insert, $O(\log n)$ לפעולה.
כשנתקלים בחיתוך, מבצעים swap ברשימת ה-y, [illegible]
ובודקים זוגות חדשים. ← יש 2 חדשים כאלה ו-1 שהתבטל, לכל היותר
סיבוכיות זמן ריצה: $O((n+k)\log n)$.
מס׳ האירועים הקריטיים $2n+k$, $O(\log n)$ זמן בכל אחד.

- עבור מקומות שונים, האלגוריתמים שונים.
ישרים: אתחול y לפי סדר שיפוע יורד ואתחול רשימת x עם החיתוכים.
מעגלים: כל כניסה/יציאה ← הוספה/מחיקה של שני חצאים.