העניין הוא שמיון והכנסה ל-x-str זה קל, אבל איך להפעיל במקביל sweep? הכל שם תלוי בהכל!

אבל יש פתרון. בעצם לא צריך למקבל את כל המימוש של ה-sweep.

מקבול:

~~אבחנה: קודם כל, בגלל שכל המעגלים באותו רדיוס, אם יש חיתוך בין שני מעגלים, אז הוא יהיה בין אלה שהצלע של המרכז לגלגל המרכז של האחר.~~
(מורית שמה לב שלא נכון)

אבחנה: כדי ~~ש~~שני מעגלים יחתכו, המרכזים צריכים להיות לא רחוקים מאד מהשני על ציר x.

$$|p_x - q_x| < \delta$$

עושים לוקליזציה לאיפה עלולים להיות נק' חיתוך, ופותרים את בעיה ונמשיך עם הפרטים בשבוע הבא.

הרצאה (2)

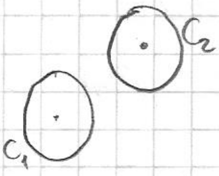
ממשיכים עם הבעיה מפעם שעברה:
מחפשים מרחק מינימלי בין n נק' במישור. פונק' ההכרעה - האם המרחק $\geq \delta$?
פותרים ע"י בדיקה אם יש חיתוך בין מעגלים ברדיוס $\delta/2$ סביב הנק' ע"י sweep בזמן $O(n\log n)$.
כדי להדגים את המימוש הפרמטרי, נתעמק יותר בבעיית מקבול ה-sweep במקרה המיוחד שלנו, אפילו שזה לא באמת לב העניין פה.

תזכורת: מסמלצים את פונק' ההכרעה על x^*, תוך כדי צבים ערכים קריטיים שבהם תשובה על שאלות מסויימות בפונק' משתנות. את התשובה לשאלות שצצות, פותרים ע"י הרצה קונקרטית של פונק' ההכרעה על הערך הקריטי שצץ.

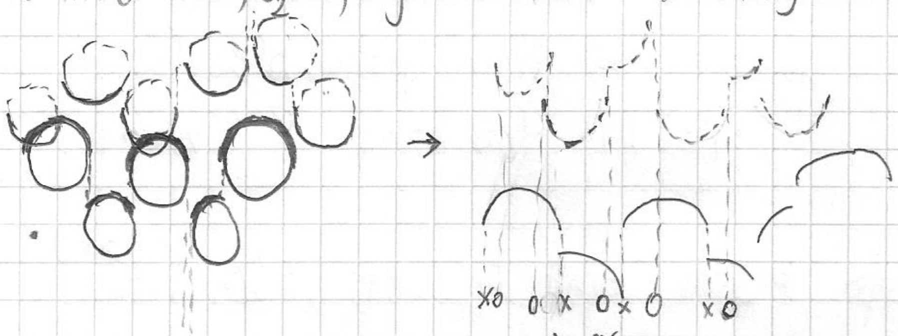
ביצוע מקבילי: לא ע"י sweep. נחלק את קבוצת המעגלים C ל-2 קבוצות C_1, C_2, בגודל $\frac{n}{2}$ כ"א (הפרד ומשול...). ברקורסיה נבדוק האם יש חיתוך במעגלי C_1 ובמעגלי C_2.

גישת הפרד ומשול נהדרת אם רוצים מקבול. השאלה מה עושים
ב-"merge", כלומר כיצד בודקים חיתוכים בין מעגלים מ-C_1 למעגלים מ-C_2.

נחלק את המעגלים כך שכל המרכזים ב-C_1 נמוכים מהמרכזים ב-C_2 (לפי y



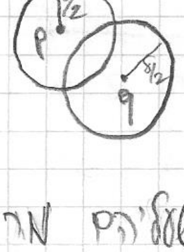
← כל מעגל ב-C_1 נמצא מתחת לכל מעגל ב-C_2, אם הם
לא נחתכים.
← לכן, מספיק לבדוק חיתוך בין המעטפת העליונה של C_1, לבין
המעטפת התחתונה של מעגלי C_2.
מה זה מעטפת - בכל x נשמור את המעגל הכי נמוך מ-C_2, והכי גבוה מ-C_1.



נניח שהרקורסיה מחזירה את המעטפת.
כל מעגל מופיע רק בקשת אחת על המעטפת.
← נמזג את הרשימות לרשימה אחת, והרי מיזוג אפשר לעשות באופן מקבילי
בעומק $O(\log n)$, ומספר ליניארי של "מעבדים" (כלומר בכל שלב במימוש, יהיו
לכל היותר מספר לינ' של שאלות.

הרשימה הממוזגת, מורכבת מאינטרוולים, ($O(n)$ אינטרוולים), בכל אינטרוול, כל
מעטפת מיוצגת ע"י מעגל יחיד, ונבדוק חיתוך בין זוג המעגלים הללו.

x o x o o o x o x x o x ---
כמובן, באינטרוולים בהם אין מעגל (כי יכולים
להיות חורים במעטפת), לא עושים כלום, זה נהדר.



המעגלים נחתכים $\Longleftrightarrow$

$$\|pq\| \le \varepsilon$$

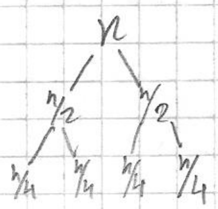
המרחקים בין מרכזי המעגלים הם הערכים הקריטיים, שעליהם מריצים
את הפרד' קונקרטית.

אם אין חיתוך, צריך לייצר מעטפת עליונה ותחתונה של C מתוך המעטפת התחתונה והעליונה של C_1 ושל C_2.

עליונה: נמזג את הסדרות של 2 המעטפות העליונות ונמצא ובכל אינטרוול יהיו ≤ 2 עיגולים פוטנציאליים. זה אפילו יותר קל, כי אם יש מעגל מ-C_1 אז הוא יהיה העליון גם אם אין אז C_2.

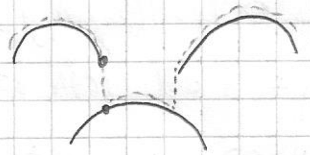
$\therefore$ העומק הכולל פה הוא $O(\log^2 n)$ כי לרקורסיה יש עומק $\log n$, ובכל שלב רקורסיבי, המיזוגים הם בעומק $O(\log n)$.

בספרות אגב, מיזוג (מוסבר איך) אפשר לבצע בעומק $O(\log\log n)$.



בכל שלב, האורך הכולל של הרשימות הוא $O(n)$, כי סכום כל רמה בעץ הוא $O(n)$.

(ניזכר) שבכל שלב, נחשב את $O(n)$ הערכים הקריטיים של ההשוואות.



דברים משתנים כאשר המעטפת משתנה, וזה קורה כאשר מעגל מסוים מפסיק להיות מעל מעגל אחר (כאשר משנים את δ).

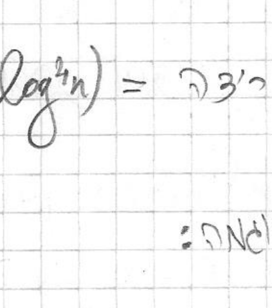
כלומר הערכים הקריטיים הם הפרשי קואורדינטת x בין מרכזי מעגלים, ודברים קורים (עולים) כאשר $\delta = p_x - q_x$.

$\Leftarrow$ $O(n\log n) \times O(\log n) \times O(\log^2 n)$: סה"כ זמן ריצה $= O(n\log^4 n)$.

קריאה לפונקציה ההכרעה | חיפוש בינארי על ערכים קריטיים | עומק מקבילי

ועוד דוגמה:

<u>מרחק האוסדורף בין קבוצות</u>



יש 2 קב' A ו-B של נקודות. $|A|=|B|=n$.

רוצים לבדוק דמיון בין A ל-B. הגדרה (חד כיווני):

$$H(A,B) = \max_{a\in A}\min_{b\in B}\|a-b\|$$

המרחק המקס' של נק' ב-A משכנה הקרובה ביותר ב-B.

איך משבים את זה ביעילות? עושים דיאג' וורוני' ל-B, מעבדים אותה לכדי מפה מישורית עבור point location, וכך אפשר לחשב את $H(A,B)$ ב-$O(n\log n)$.

מתי זה עשוי להיות מעניין? נניח ש-2 הקב' נתונות במע' צירים שונות, ורוצים להשוות ביניהם. אז היינו רוצים לחשב מרחק האוסדורף תחת טרנספורמציות. אנחנו נתמקד <u>במרחק האוסדורף תחת הזזות</u>: מותר להזיז את A, ומחפשים הזזה t שתביא למינימום את המרחק $H(A+t,B)$.

נעבר כאן בניסוח פרמטרי.

<u>פונק' ההכרעה</u>: נתון δ. האם קיים t כך ש- $H(A+t,B)\leq\delta$.

השאלה האם $H(A+t,B)\leq\delta$ זה בעצם האם המרחק של כל נק' מ-A לשכנה הקרוב ביותר ב-B $\leq\delta$.

סביב כל $b\in B$ נצייר עיגול $D_\delta(b)$ ברדיוס δ, וכעת השאלה היא שקולה להאם כל נק' מ-A נמצאת בעיגול אחד לפחות, כלומר האם A מוכלת באיחוד העיגולים U_δ.

זה עבור t ספציפי. אנחנו רוצים לדעת האם קיימת הזזה t כך ש- $A+t\subseteq U_\delta$.

עבור $a\in A$ ההזזות שמקיימות $a+t\in U_\delta$, הן הקב' $U_\delta-a$.

כלומר לקחת את קבוצת העיגולים ולהזיז אותה ב-$-a$.

נעשה זאת לכל a, וכעת שואלים האם קיימת t משותפת לכל ה-$U_\delta-a$, כלומר האם $\bigcap_{a\in A}(U_\delta-a)\neq\emptyset$.

(מזיזים את ענן העיגולים n פעמים, לכל נק' ב-a. אם חיתוך כל העננים באמצע אינו ריק, אז זה יתן t כך שכל נק' a תהיה בתוך ענן כלשהו).

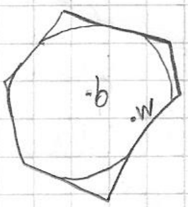
<u>הערה</u>: סיבוכיות U_δ היא $O(n)$.

היות והעיגולים שווי רדיוס זה בכלל קל להוכיח, אבל זה נכון גם לכל קב' של עיגולים.

הוכחה:

נבנה את דיאג' וורונוי של B, ונשים שבתא וורונוי של איזשהו מרכז b,

$$U_\delta \cap V(b) = D_\delta(b) \cap V(b)$$



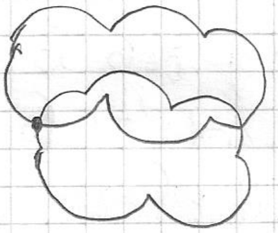
אם w היא בתוך האיחוד אז מרחקה מנק' כלשהיא קטן מ-δ, והרי b היא הכי קרובה אליה אם w בתא וורונוי של b, אז המרחק בין w ל-b קטן מ-δ, והוא בדיסק שלה.

במילים אחרות: $w \in U_\delta \cap V_b \Rightarrow \exists b'\ wb' \le \delta \Rightarrow wb \le \delta = w \in D_\delta(b)$

לכן, גם לכל עותק $U_\delta - a$ יש סיבוכיות לינארית.

לכל 2 עותקים $U_\delta - a, U_\delta - a'$, החיתוך הוא גם מסיבוכיות לינארית.

הסיבה: קדקדי החיתוך הם גם קדקדי האיחוד של 2n המעגלים האלו (נכון רק ל-2 מעגלים! עובד רק בזוגות). והרי איחוד הוא לינארי...



$\Leftarrow$ לכן החיתוך של m עותקים כאלו, סיבוכיותו $\le \binom{m}{2} \times$ סיבוכיות חיתוך/איחוד של 2 עותקים.
$\Rightarrow O(m^2 n)$

ולכן סיבוכיות החיתוך היא $O(n^3)$.

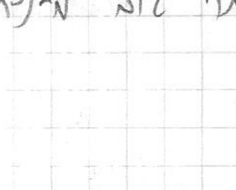
אבל, הרי לא באמת מעניין אותנו החיתוך של כל העותקים. אנו רק רוצים לדעת האם החיתוך לא ריק... אבל לא יודעים משהו יותר טוב מ-$O(n^3)$.

נבצע D&C (לחישוב:) $K = \bigcap_{a \in A} (U_\delta - a)$:

$n/2$ עותקים, ומחשבים את חיתוכם, K_1,

$n/2$ עותקים, " " " K_2,

ומחשבים $K_1 \cap K_2$.



את זה נעשה ע"י sweep. כל נק' חיתוך בין שפת K_1 לשפת K_2 היא קדקד של k.

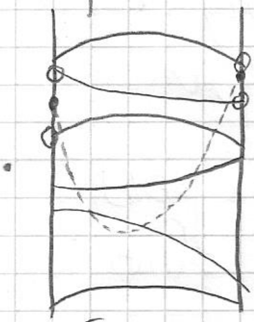
כמה זה עולה לנו?

$$\underbrace{T(m, n)}_{m:\ \text{עננים},\ n:\ \text{כמה בענן}} = 2T(m/2, n) + O(m^2 n \log n) \Rightarrow T(m,n) = O(m^2 n \log n)$$

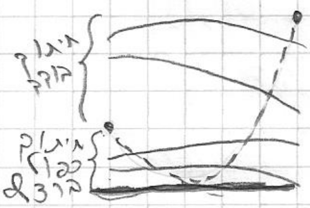
$$T(n) \overset{\downarrow}{=} O(n^3 \log n)$$

השאלה היא איך ממקבילים? למצוא את זה מאוד קל (בהנחה לו).
במקום ה-sweep נעשה משהו דומה למקרה שראינו קודם.
המקבל הוא טבעי עבור ה-D&C. הוא יותר בעייתי עבור גישה אחרת.
אם כן, יש לנו $O(m^2n)$ קשתות מעגלים אדומים ו-$O(m^2n)$ קשתות מעגלים כחולים.

נניח שכל הקשתות הן מונוטוניות ב-x (ואם לא נשבור אותן ל-2 *).



נניח שיש לנו "פס" ובו אין חיתוכים בין הקשתות האדומות שחוצות אותו (הן זרות).
כעת, נכניס קשת כחולה, ורוצים לדעת אילו מעגלים היא חותכת.
ע"י חיפוש בינארי נמצא בין אילו 2 קשתות, על כל גבול של הפס, הקשת הכחולה נמצאת.

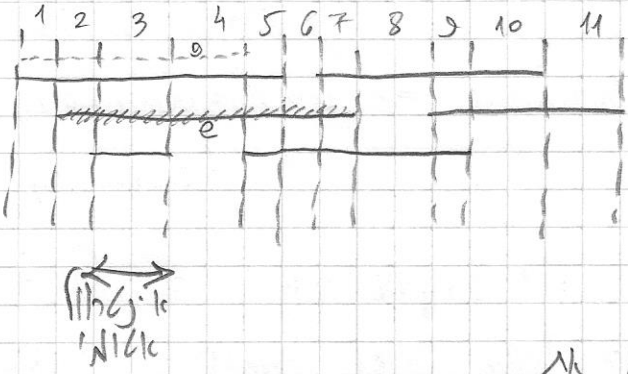


אז מחלקים ל-2 רצפים לפי גובה חיתוך הקשת הכחולה עם גבולות הפס, ואת הרצפים אפשר למצוא ע"י חיפוש בינארי ונקבל אפילו את סדר החיתוכים בכל רצף, כי כל האדומות לא נחתכות. ברצף העליון יש חיתוך בודד (אם מתחילים מעל נק' הקצה השמאלית של הקשת הכחולה (אם היא יותר נמוכה)), וברצף התחתון יש 2 נק'.

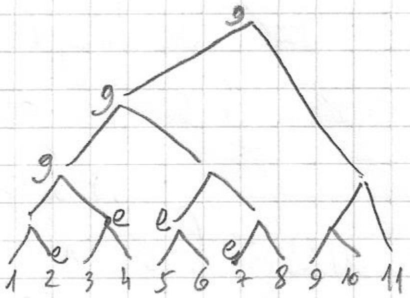
אם הפס נראה כך, אחלה. אם לא, איך מביאים אותו למצב הזה?
אז צריך עוד טכניקה בסיסית שלרוב מלמדים בקורס הבסיסי:

Segment tree

ניקח את הקשתות האדומות, נטיל אותן על ציר ה-x, ונקבל $O(m^2n)$ קטעים. נאמן אותם ב-segment tree



כל האינטרוולים האטומיים
נמצאים בעלים.
כל קטע s מאוחסן בצמתים
v כך ש-s מכסה את הטווח
של תת העץ של v, אבל לא את
זה של האב של v.



$\Rightarrow$ צריך $n\log n$ מקום
ל-n קטעים.

כי כל קטע מופיע ב $\leq 2\log n$ צמתים,
שניים בכל רמה (בגלל הקצוות שלו).

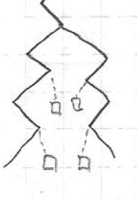
נוסיף את הקשתות הכחולות באותו אופן, אבל גם נשלח כל קשת
כחולה לכל האבות הקדמוניים של כל הצמתים בהם היא מאוחסנת.

כך נקבל את הפסים הרצויים.

נו, יופי. ואת כל העניינים האלו קל למקבל, שלא כמו sweep. כל מה שיש
לנו פה זה כל מיני מיונים, חיפושים בינאריים, ו-D&C שכבר ראינו
שנתנה למקבול.

<u>זמן ריצה</u>: מורכב מ-$n\log^3 n$ של פרוצ' ההכרעה, עם עוד כמה לוגים.

סגם קצת ההכרעה לגבי הסגמנט-טרי. מאחסנים ב-seg-tree גם קשתות כחולות
וגם אדומות. את האדומות מאחסנים לפי הכלל, ואת הכחולות גם לפי הכלל וגם
שולחים לכל האבות הקדמוניים. כמו כן, חוזרים על הפעולה בהיפוך
3 פעמים.



צריך להשתכנע שזה לא פוגע בסיבוכיות, כי כל
"השליחות" למעלה לאבות הקדמוניים מתקפלים למסלול
בודד בסופו של דבר, שהם באורך $n\log n$... ואין יותר
מדי שכפולים...

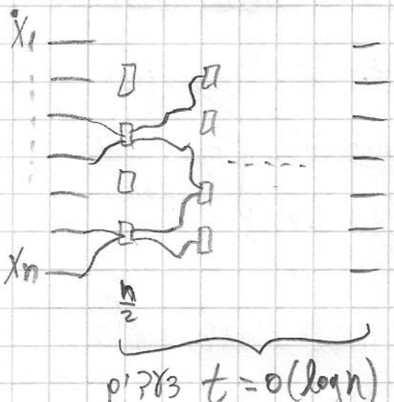
נזכור שמפסיקים נק' חיתוך כמות אקומת, וכל נק' כזו קורית באיזשהו אינטרוול אטומי בסופו של דבר.

כעת לדוגמאות אחרת, קצת יותר קדרדרבת, אבל לפני זה הערה לגבי חסכון של לוג. נחזור לרגע לדוגמה של slope selection. כזכור, ראינו $O(n\log^3 n)$.

* log מיון.
* log חיפוש בינארי.
* log מפרוצ' ההכרעה.

הבה נחסוך log, ע"י "השיפור של Cole".

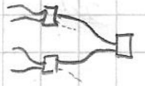
רוצים להמנע מחיפוש בינארי של הערכים הקריטיים בכל שלב מקבילי.
הטכניקה עובדת עבור בעיות שפונק' ההכרעה בהן מבוססת על מיון, ומשתמשת במימוש מקבילי המבוסס על רשת מיון.



הטריק של Cole אומר, במקום לבצע את כל השלבים של החיפוש הבינארי, בואו נבצע רק את השלב הראשון.
נקבל מצב שאחרי $\frac{n}{2}$ שערים $\rightarrow$ $n/2$ ערכים קריטיים.
במקום חיפוש בינארי מלא, נשווה רק עם החציון.

$x_1, x_2, \ldots, \mu, \ldots x_{n/2}$

חציון



נשאל עבור x^*: μ כלומר $x^* \lesseqgtr \mu$?
זה יתן לנו תשובה לחצי מההשוואות!

אח"כ עוברים לשלב הבא של השערים וכעת צריך להיזהר כי עלול להיות מצב שמתמודדים עם שער שלא בדקנו את הקלטים שנכנסים אליו בשלב הקודם.
כדי שלא ייווצרו מצבים כאלו, נעזר בחציון משוקלל.

במילים של מיכה: בכל שלב יש שערים פעילים: יודעים מהם 2 הקלטים וצריך/אפשר לבצע את ההשוואה.
לכל שער פעיל בשכבה ה-j של הרשת, ניתן משקל $\frac{1}{4^j}$ $(j=0,1,\ldots)$

נסמן ב-W את המשקל הכללי של השערים הפעילים בצעד אחד, וניקח את הערכים הקריטיים של ההשוואות שלהם: $x_1, x_2, \ldots x_m$, ונחשב חציון משוקלל שלהם כך שמשקל כל הערכים שמתחתיו $\ge \frac{W}{2}$ ומעליו $\ge \frac{W}{2}$. נסמן אותו בתור λ^*, ונקרא לפונק' ההכרעה עם λ^*, כלומר $x^* : \lambda^*$.

ההכרעה פותרת את מחצית ההשוואות, כלומר בסה"כ פתרנו השוואות במשקל כולל $\ge \frac{W}{2}$. המשקל הכולל של השערים הפעילים קטן בלפחות $\frac{W}{2}$.

כל שער בשכבה j כשהוא שנפתר/הפסיק להיות פעיל יכול להפוך ≤ 2 שערים בשכבה הבאה לפעילים.

כלומר החלפנו שער שמשקלו $\frac{1}{4^j}$ בלכל היותר 2 שערים שמשקלם $2 \cdot \frac{1}{4^{j+1}} = \frac{1}{2} \cdot \frac{1}{4^j}$.

המשקל הכללי ירד ב- $\frac{W}{2}$ ועלה ב- $\frac{W}{4}$, אז הוא ירד ב- $\frac{W}{4}$, ולכן המשקל הכללי בשלב הבא $\le \frac{3}{4}W$.

ולכן, אחרי k שלבים, המשקל הכללי של הצמתים הפעילים $\le \frac{n}{2} \cdot \left(\frac{3}{4}\right)^k$,

אם עדיין יש צומת פעיל, משקלו $\ge \frac{1}{4^t}$

$$\frac{1}{4^t} \le \frac{n}{2}\left(\frac{3}{4}\right)^k \Rightarrow k = O(t+\log n)$$

כלומר אחרי שעוברים את המספר הזה, אין כבר צמתים פעילים, ובעצם רשת המיון סיימה את עבודתה.

אם $t=O(\log n) \Rightarrow k=O(\log n)$.

אז יש לנו $O(\log n)$ שלבים שבכל אחד עושים רק קריאה אחת לפרוצ' ההכרעה. כך מוסיפים גורם log וזה נותן ל-slope selection $O(n^2 \log n)$.

קשה לדמיין איך אפשר להמנע מקריאה לפונק' ההכרעה פחות מ-$O(\log n)$ פעמים.

כדי להוריד עוד log צריך לנצל רעיונות אחרים, למשל אם מריצים את פרוצ' ההכרעה מאוד רחוק מאיזור העניין, אז אפשר להעריך את התשובה עד שגיאה מסוימת. כל פעם השגיאה תגדל, ורק בחלק מהמקרים נרצה לחשב יותר מדויק וכו', וכך חוסכים איכשהו עוד log.