26.1.09

גאומטריה חישובית

נרצה להכליל את דיאגראמות וורוני...

נתונים - אתרים   - מימד

- מטריקה   - k השכנים הקרובים ביותר

דיאגרמה במימד d מתנהגת כמו מעטפות תחתונות של על-מישורים ב $d+1$ מימדים. כלומר כמו קמור תחתון ב $d+1$ מימדים.

לכן, סיבוכיות הדיאגרמה היא $O(n^{\lfloor (d+1)/2 \rfloor}) = O(n^{\lceil d/2 \rceil})$

וניתן לחשב אותה ע"י חישוב קמור בזמן רנדומי.

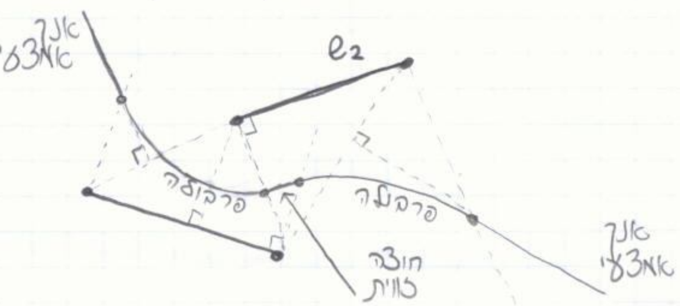
מה קורה אם אתרים שאינם נקודות, למשל קטעים במישור (מטריקה אוקלידית)? ראשית, האתרים זרים זה לזה ולכן S היא קבוצה של n קטעים זרים במישור.

$Vor(S)$ = חלוקת המישור לתאים כך שהתא של קטע e הוא G הנקודות שהאתר הקרוב להן ביותר הוא e, כאשר

$$d(x,e) = \min_{y \in e} d(x,y)$$

מה ניתן להגיד על ה bisector בין 2 קטעים e_1, e_2?

דוגמה -



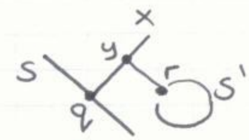
כלומר ה bisector הוא צירוף של פרבולות, אנכים אמצעיים וחוצי זווית.

- אם כך, תאי וורוני אינם בהכרח קמורים. לעומת זאת, הם כן star-shaped (כוכביים) ביחס לאתרים שלהם:

לכל אתר s ולכל $x \in V(s)$, הקטע המחבר את x לנקודה הקרובה ביותר על s מוכל כולו ב $V(s)$.

הסבר -

$y \notin V(s)$



נניח

ואז $yr < d(y,s) = yq$

נכון אם האתרים קמורים

$\Rightarrow xr \le xy + yr < xy + yq = xq$

$\Rightarrow d(x,s') \le xr < xq$

בסתירה לכך ש $x \in V(s)$

· בפרט, תאי וורונוי הם קשירים (לאתרים קמורים זרים).

אם בוורונוי יש n פאות ובתנאי שלכל קודקוד וורונוי יש דרגה ≥ 3

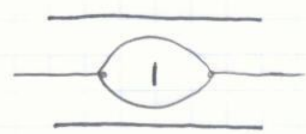
(וזה נכון) אז מתקיימת נוסחת אוילר $E \le 3n-6$

$. V \le 2n-4$

· במקרה של אתרים נקודתיים, ל 3 תאי וורונוי הייתה לכל היותר נקודה אחת משותפת.

באופן כללי, יתכנו 2 נקודות משותפות אך לא יותר

דוגמא -



(ולכן יתכנו 2 נקודות משותפות)

הסבר -

למה לא יתכנו יותר מ 2? אילו היו 3 אתרים S_1, S_2, S_3 עם 3 קודקודי וורונוי משותפים a, b, c היינו יכולים לבנות ציור מישורי של הגרף $K_{3,3}$ וזו סתירה (לקמורות)

· הבחנה נוספת היא של bisector הוא קו חלק (כלומר פונקציה גזירה)

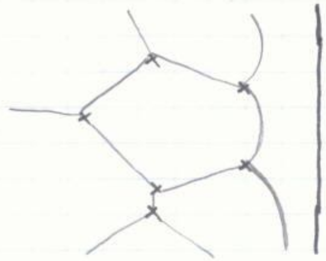
ואחרי כל זה, נתאר כיצד ניתן לחשב את הדיאגרמה (של האתרים שהם קטעים):

נעשה בעזרת ש"י sweep (Fortune).

במקרה נקודתיים, מה שהיינו עושים זה

ומתחזקים את $Vor(S_L \cup \{l\})$ בכל אירוע ברשימת ה x.

איך נראה $Vor(S_L \cup \{l\})$? כבר הזכרנו שה bisector בין l לנק' a הוא פרבולה עם a כמוקד ו l כמדריך. כל הפרבולות מונוטוניות ב y והשפה של $V(l)$ היא חזית פרבולית.



ניתן גם להראות שמס' הקשתות בחזית הוא $\leq 2m-1$ (כאשר m הוא $|S_L|$).

ומה קורה ל $Vor(S_L \cup \{l\})$ כאשר l זז ימינה? כל הקשתות חוץ מהפרבולות הן קשתות "אמיתיות" של $Vor(S)$ ורק החזית משתנה.

יכולים להיות 2 סוגים של אירועים קריטיים

(1) קשת פרבולית מתכווצת לנקודה ושתי הקשתות השכנות שלה נהיות שכנות בחזית (מתגלה קודקוד וורוני)

(2) l עובר דרך אתר חדש

אנו נחזיק y-str שהוא החזית הפרבולית (קודקודים שבינהם קשתות פרבוליות) לפי סדר y

x-str שהוא האירועים הקריטיים, כאשר אירועים מסוג (2) ידועים מראש ואירועים מסוג (1) מתעדכנים.

כל שלשה של קשתות עוקבות בחזית בודקת מתי הקשת האמצעית נעלמת (= מהו מיקום l בו הוא שווה מרחק לקודקוד הוורוני המתאים)

כל שינוי דיסקרטי של החזית הפרבולית נחפש שלשות עוקבות חדשות ונמצא עבורן את האירועים הקריטיים המתאימים.

באירוע אתר (2) נוספת קשת פרבולית ומתפצלת קשת קיימת לשני תתי קשתות, 2 קודקודים חדשים, ונוצרת חתיכה קטנה מקשת וורוני אמיתית.

האלגוריתם ירוץ בזמן $O(n\log n)$

הסבר -

צריך להוכיח שמס' האירועים הקריטיים הוא $O(n)$. ואכן, אירוע קריטי נוצר כששינוי אמיתי בדיאגרמה קורה (מתגלה קודקוד / נוסף אתר) וכל שינוי אמיתי יוצר לכל היותר שתי שלשות חדשות עוקבות,

ועבור אירועים שהם קטעים נעשה אותו דבר (רק התיאור הוא יותר מסובך)

[illegible] כאן דיאגרמות וורוני.

Range Searching:

נתונה קבוצת נקודות P במישור. כמו כן, נתון אוסף של טווחים אפשריים, כאשר כל טווח הוא צורה מסוג פשוט יחסית (מלבן מקביל לצירים, עיגול, משולש, חצי מישור, ...)

המטרה היא עיבוד מוקדם של P למבנה נתונים שיאפשר לענות ביעילות על שאילתות טווח מהסוג

"בהינתן טווח R מהאוסף הספציפי, יש לחשב כמה נקודות של P יש ב R" (או לדווח אותן, או לבדוק אם R ריק)

בניגוד לבעיות קודמות שראינו, במקרה זה קיים trade-off בין $S(n)$, (זכרון ! $P(n)$.

נצמצם את הדיון שלנו לטווחים שהם מלבנים מקבילים לצירים.

נתחיל ממקרה עוד יותר פשוט - המקרה החד-מימדי - בהינתן קטע $[a,b]$ לדווח את קבוצת הנקודות שנמצאות בין a ל b:

נחזיק את הנקודות בעץ חיפוש בינארי מאוזן (ממוינות לפי x).

נחפש את a,b בעץ ונקבל 2 מסלולי חיפוש.
למעשה, נקבל אוסף של תתי-עצים זרים התלויים
מתחת למסלולי החיפוש בניהם. יש $\leq 2\log n$ תתי עצים כאלה
והם יהיו הייצוג שלנו

קיבלנו $Q(n)=O(\log n)$ $S(n)=O(n)$ $P(n)=O(n\log n)$

נחזור לדון במקרה הדו-מימדי...

נבנה עץ T כנ"ל (ראשון) עבור קואורדינטת x של הנקודות
לכל תת-עץ T_u ששורשו בצומת u כלשהו, נבנה עץ שניוני על
הנקודות של T_u ממוינות לפי y.

כדי לענות על שאילתה, נחפש את $[a,b]$ בעץ הראשון ונקבל $\leq 2\log n$
תתי עצים.

נלך לכל תת עץ שניוני המתאים לתתי העצים הנ"ל ונחפש בו את $[c,d]$

ומה הסיבוכיות?

$$Q(n)=O(\log^2 n)$$
$$S(n)=O(n\log n)$$
$$P(n)=O(n\log n)$$

ניתן להכליל אלגוריתם זה לכל מימד נקבל

$$Q(n)=O(\log^d n),\quad S(n)=O(n\log^{d-1} n),\quad P(n)=O(n\log^{d-1} n)$$

הערה-

ניתן לשפר את $Q(n)$ במישור ל $O(\log n)$

ה"טריק" הוא שלכל קודקוד u בעץ הראשון שבניו הם w, v נחזיק
מצביע לנקודה הקרובה ביותר ברשימת v ומצביע לנקודה הקרובה ביותר
ברשימת w

נעביר את הדיון לטווחים שהם חצאי מישור...

נשתמש בדואליות- הנק' עוברות לישרים והשאילתה היא נקודה q כך
שצריך למצוא כמה ישרים מתחת ל q.

וזה הופך לבעיית point-location של q במערך = המפה המישורית שהישרים יוצרים.
יהיו לנו $O(n^2)$ אירועים ולכן

$Q(n) = O(\log n)$
$S(n) = O(n^2)$
$P(n) = O(n^2 \log n)$

אם היינו דורשים $S(n) = O(n)$ אז ניתן לספור עם
$Q(n) \sim O(\sqrt{n})$