3.11.08

גאומטריה חישובית

המטרה לפתור בעיות עם מבנה גאומטרי (וקלט, n, גדול) בצורה אלגוריתמית יעילה

דוגמה לבעיה כנ"ל-

- נתונות n נקודות במישור. יש לחשב את המרחק הקצר ביותר בין 2 נקודות.
- נתונים n קטעים במישור. יש לבדוק האם קיים חיתוך בין 2 קטעים / יש לחשב את כל זוגות הקטעים הנחתכים.

ניתן לראות שלא מדובר דווקא על בעיות np-complete אך גם במקרים כמו הדוגמאות האחרונות, יש פתרון ברור של $O(n^2)$ נרצה למצוא פתרון יותר יעיל.

לפני שנמשיך, ננסה לאפיין קצת את הבעיות איתן נתמודד

- הקלט הוא מספרים ממשיים (וכאן גם יש סוגיה של דיוק)
- הפלט הוא מספרים ממשיים או מידע דיסקרטי אחר (ושוב חוזרת בעיית הדיוק במהלך החישובים)

נשים לב שעקב הסוגיות לעיל ייתכן שגם בבעיות שנראות פשוטות (כמו הדוגמה הראשונה) האלגוריתם לא יעבוד (ייתכן שנחזיר זוג נקודות שהמרחק ביניהם הוא רק קרוב לקטן ביותר). מעבר לזה, אם ניקח, לדוגמה, על בעיה כגון חישוב השטח מסביב לנקודה בציור הבא , אז ייתכן שבגלל טעות בדיוק (האם הקווים נחתכים) נחזיר תשובה מאוד שונה או לא תשובה בכלל.

ישנו פתרון לבעיה זו, חישוב מדוייק במספרים ממשיים, אך זהו תחום אחר שלא נתעסק בו בקורס זה

בעיה נוספת שעליה יש לשים לב היא קלט במצב מנוון (סוג של מקרי קצה). לדוגמה, בבעיית חיתוך הקטעים נצטרך לטפל גם במקרים  .

במרבית הקורס אנחנו נניח מצב כללי (general position) ונתעלם גם מבעיה זו.

בגלל ההנחה שלנו של חישוב מדויק אנחנו ננסה בקורס להשתמש רק בכלים אלגבריים (ולא, לדוגמה, בטריגונומטריה) ועבורם נניח זמן קבוע לכל פעולה (ואפילו לפעולות "אסורות" כמו הוצאת שורש)

<u>חישוב קמור של קב' נקודות</u> (convex hull)

קב' קמורה במישור-

C קמורה אם"ם לכל שתי נקודות $p,q \in C$ הקטע הישר המחבר אותן מקיים $C \supset [p,q]$

קמור של קבוצה-

בהנתן קב' P של נקודות במישור, הקמור של P, שסימון $CH(P)$ או $conv(P)$, הוא הקב' הקמורה המינימלית שמכילה את P.

כלומר, חיתוך כל הקבוצות הקמורות המכילות את P.

תכונת ההפרדה-

אם C קבוצה קמורה ו $p \notin C$ אזי קיים ישר שמפריד בין p ו C

טענה-

קמור של P הינו חיתוך כל חצאי המישורים המכילים את P

(כאשר אנחנו מתעלמים מהניואנס של קבוצות פתוחות/סגורות)

הוכחה-

ראשית, ברור כי החיתוך של כל הקבוצות הקמורות $\subseteq$ חיתוך כל חצאי המישורים.

להפך, אם q בחיתוך חצאי המישורים אז היא בחיתוך של כל הקבוצות הקמורות, כי אם לא אז יש קבוצה קמורה $C \supseteq P$ שלא מכילה את q.. לכן קיים ישר l שמפריד בין q ל C ולכן יש חצי מישור שמכיל את C (ולכן גם את P) אבל לא את q. קיבלנו סתירה ולכן $q \in$ חיתוך כל הקבוצות הקמורות.

מסקנה -

כדי להבין מהו הקמור מספיק לחתוך את כל חצאי המישורים החסומים ע"י ישרים שעוברים דרך זוג נקודות של P

נחזור לחישוב הקמור...

נניח $|P|=n$

לפי המסקנה ניתן להסיק בקלות לכך שעבור כל נקודה יש להסתכל לכל היותר על 2 ישרים.

לכן מספר הישרים/חצאי מישור המגדירים את הקמור $\leq n$.

נבחין בין 2 סוגים של נקודות ב P

* נקודות פנימיות - בפנים של הקמור
* נקודות שפה - על שפת הקמור

בהינתן מצב כללי, כל קודקודים של הקמור ($\equiv$ נק' קיצוניות) או פנימיות.

תחת הנחה זו מס' הישרים הוא ~~[illegible]~~ כמס' נקודות השפה.

אבחנה נוספת תחת הנחה זו היא שהקמור הוא מצולע קמור בעל $\leq n$ צלעות שקודקודיו הם נקודות ב P (ובכלליות, $CH(P)$ הוא מצולע שקודקודיו חלק מ P).

ישר תומך -

ישר תומך בקמור (או באופן כללי בקבוצה קמורה) הוא ישר שעובר דרך נק' של הקבוצה אך כל הקבוצה נמצאת בצד אחד שלו

(אוו עליו)

בחזרה לחישוב הקמור...

קלט: קבוצה P של n נקודות (במצב כללי) הנתונות ע"י הקואורדינטות שלהן

פלט: 1. רשימה מעגלית של האינדקסים של קדקדי הקמור, בסדר בו הם מופיעים על שפת הקמור

2. סתם רשימה לא מסודרת של הקדקדים

אלגוריתם טריוויאלי -

לכל זוג נקודות $a,b\in P$ נבדוק אם ab צלע של הקמור (כלומר נבדוק אם כל שאר הנקודות של P נמצאות באותו צד של ab).

סיבוכיות אלגו' זה היא $O(n^3)$

הערות -

· אבל לא ברור איך נבדוק אם הנקודות באותו צד?

נניח שאני רוצים לסדר את הקודקודים בכיוון הפוך לכיוון השעון (counterclockwise $\equiv$ CCW $\equiv$) ואז נבדוק אם הנקודות נמצאות משמאל לישר $\vec{l_{ab}}$.

· ואיך נעשה את זה?

הפעולה הפרימיטיבית: נתונה שלשה סדורה של נק' a,b,c. האם c משמאל לישר $\vec{l_{ab}}$.

נעשה זאת ע"י בדיקת פנייה שמאלה (פה בד"כ היינו עוצרים כי ברור שפעולה זו ניתנת לביצוע ב $O(1)$, אבל הפעם נפרט יותר):

משוואת l_{ab} היא $y-y_a=\frac{y_b-y_a}{x_b-x_a}(x-x_a)$. לכן (...) התשובה למיקום c ניתן לפי הסימן של $(x_b-x_a)(y_c-y_a)-(x_c-x_a)(y_b-y_a)$

ולכן לפי הסימן של הדטרמיננטה
$$\begin{vmatrix} 1 & x_a & y_a \\ 1 & x_b & y_b \\ 1 & x_c & y_c \end{vmatrix}$$

כך להבין זאת ניתן לראות ש $= \begin{vmatrix} x_b-x_a & y_b-y_a \\ x_c-x_a & y_c-y_a \end{vmatrix}$

כלומר נקבע לפי $\vec{ab} \times \vec{ac}$ (ולכן חיובי זה שמאלה)

אלגוריתם עטיפת מתנה -

נבנה את צלעות הקמור אחת אחרי השנייה:

- נמצא קודקוד ראשון של הקמור (למשל הנק' עם ה x המקסימלי) - q_1
- נעבור על כל שאר הנקודות ונחפש זו שיוצרת את הזווית הכי קטנה - בכל רגע יש לנו מועמדת q' וכאשר נקודה חדשה, q, מופיעה נבדוק אם $q_1 q' q$ היא פנייה שמאלה. אם כן q' נשארת המועמדת ואם לא q מחליפה אותה. בסוף קבלנו את הצלע הראשונה של הקמור - $q_1 q_2$
- נחזור על התהליך עד שנמצא צלע המסתיימת ב q_1.

סיבוכיות האלגוריתם היא $O(n^2)$, או ליתר דיוק $O(n \cdot H)$ כאשר H מס' צלעות הקמור

האלגוריתם של Graham -

(נקרא גם Graham walk או Graham scan)

הרעיון הוא לבנות את הקמור באופן אינקרמנטלי, כלומר להוסיף את הנקודות אחת אחרי השנייה ולעדכן את הקמור לאחר כל הוספה. למעשה, נחשב רק את הקמור העליון, כלומר את החלק של הקמור שמתחיל בנק' הימנית ביותר, מסתיים בנק' השמאלית ביותר ומכיל את הצלעות שאין נקודות מעליהן. בעזרת הכלים נוכל באותה דרך לחשב גם את הקמור התחתון.

- נמיין את הנק' לפי קואורדינטת ה x מימין לשמאל - $q_1 \dots q_n$
- נחזיק מחסנית שתחזיק את קודקודי הקמור העליון הנוכחי כך שבראש המחסנית נמצא הקודקוד הכי שמאלי

נעשה באמת לפי'

```
push (q1)
push (q2)
for j = 3 to n
    q` = הנק' השנייה במחסנית
    q'' = הנק' בראש המחסנית
    while q`q''qj is right-turn
        pop (q'')
        q` = הנק' השנייה במחסנית
        q'' = הנק' בראש המחסנית
    push (qj)
```