22.12.08

גאומטריה חישובית

נכלל את בעיית איתור הנקודה, וכעת נניח שהמפה M היא כזאת שכל הפאות הן משולשים (זה כלל כי פאות מצולעות ניתן לחלק למשולשים ובמקרה של קווים ישרים ניתן לעשות פירוק טרפזי ואז לפרק למשולשים)...

הרעיון יהיה לבנות סדרה של מפות $M=M_0, M_1, M_2, \dots, M_k$ (כאשר $k=O(\log n)$) כך שכל מפה יותר "גסה" מקודמתה (כלומר יותר פשוטה, פחות פאות) וכך ש-M_k משולש (או מפה של סיבוכיות קבועה אחרת).

איתור נקודה q יתבצע מהסוף - נמצא את הפאה של M_k שמכילה את q, ובכל צעד, בהינתן הפאה f_i של M_j שמכילה את q נמצא את הפאה f_{j-1} של M_{j-1} שמכילה את q בזמן קבוע.

במבנה הנתונים, כל פאה של M_j תצביע על $O(1)$ פאות של M_{j-1} ש"מרכיבות" אותה.

קל לראות שאם אכן תהיה לנו סדרת מפות שמתנהגת כך אזי עלות חיפוש תהיה $Q(n)=O(\log n)$.

לכן, הבעיה כאן היא איך לבנות את $\{M_i\}$. נראה כיצד בונים את M_1 מ-M_0 ואת זה ניתן להכליל לבניית כל M_j מ-M_{j-1}:

ראשית, נבנה קבוצה V_0 גדולה של קודקודים ב"ת (אף צמד לא מחובר בקשת). הכוונה ב"גדולה" היא קבוצה שגודלה לינארי במספר הקודקודים. אמנם הבעיה של מציאת קבוצה ב"ת מקסימלית היא NP-קשה, אבל מציאת קבוצה "גדולה" היא בעיה פתירה. נדרוש גם $d(v)\leq c$ לכל $v \in V_0$.

שנית, לכל קודקוד $v \in V_0$, נמחק את v ואת הצלעות ב-M_0 שיוצאות ממנו, ונקבל "חור" של v - אותו נפרק למשולשים.

דוגמה:

בגלל בחירת הקודקודים, החורים זרים זה לזה וכל אחד מהם
מורכב ממס' קבוע של קשתות.
לכן, ניתן בזמן קבוע לפרק חור למס' קבוע של משולשים כך שכל
פאה חדשה בחור תצביע על כל הפאות הישנות בחור ($O(1)$).
נשאר להסביר איך בונים את V_0...

יהיו $v_1, v_2, \dots, v_n$ כל קודקודי M_0. לפי נוסחת אוילר נקבל

$$\sum_{i=1}^{n} d(v_i) = 2E \le 2(3n-6) = 6n-12$$

$$\Rightarrow \overline{d(M_0)} \le \frac{6n-12}{n} = 6 - \frac{12}{n} < 6$$

ולכן קיימים לפחות $\frac{n}{2}$ קודקודים (בה"כ $v_1, \dots, v_{\frac{n}{2}}$) כך ש
$1 \le i \le \frac{n}{2}$ $d(v_i) \le 11$.

נעבור על $\{v_i\}_1^{n/2}$ לפי הסדר. כל קודקוד שנתקל בו נוסיף ל V_0
ונמחק מהרשימה את כל שכניו.
כל צעד מוחק לכל היותר 12 צמתים מהרשימה ולכן $|V_0| \ge \frac{n}{24}$
(לינארי ב n). כמו כן, ברור ש V_0 קבוצה ב"ת של צמתים
שדרגתם ≤ 11

לפי הבנייה הנ"ל, ב M_1 יש $\le \frac{23}{24}n$ קודקודים, ולכן, אם נמשיך
באותו אופן, התהליך יסתיים אחרי $O(\log n)$ מפות. את המספר
המדויק ניתן לחשב לפי $n \cdot (23/24)^k \le 1$

$$\Rightarrow k = \log n / \log \frac{24}{23}$$

כמה זמן יקח לבנות את המפות?

בניית M_1 תיקח $O(n)$ ולכן $\le cn$ עבור c כלשהו. לכן, סך כל
בניית המפות הוא (עפ"י טור גאומטרי)

$$T \le cn + c \cdot \frac{23}{24} n + c \cdot \left(\frac{23}{24}\right)^2 n + \dots + c\left(\frac{23}{24}\right)^k n \le 24 \cdot c \cdot n$$

ומה עם סיבוכיות זכרון?

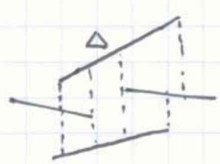
מאותה סיבה, הזכרון הכולל לכל המפות הוא $O(n)$

הערה-

הסיבה שהצלחנו לבנות זמן עבוד מוקדם של $O(n)$ ולא $O(n\log n)$
כאן מקודם היא שהמפה כבר מחולקת למשולשים.

נביא פתרון נוסף לבעיית איתור הנקודה (אלג' רנדומי אינקרמנטלי)...
הרעיון הוא לבנות את M ע"י הוספה של הקשתות שלה אחת אחרי השניה, בסדר אקראי, ותוך כדי לתחזק את הפירוק הטרפזי של המפה אחרי כל הוספה באופן שכל המבנה יתמוך ב point-location יעיל, כלומר, כל נקודת שאילתה תדע באיזה טרפז היא נמצאת.

דוגמה-

אם המפה הייתה 

ואח"כ היא 

אז כל נקודה q שכבר "יודעת" שהיא ב Δ צריכה רק לבדוק האם היא ב Δ_1 או Δ_2 ואת זה ניתן לבצע בקלות ע"י בדיקה האם q מעל/מתחת לצלע החדשה

כדי לא להגיע למצב שבו נצטרך כל שאילתה לעבור על כל השלבים (לינארי) נרצה שכל שאילתה נעבור רק במקומות בהם יש שינוי, כלומר בהם הטרפז שבו q נמצא נהרס. נראה אח"כ שבגלל האקראיות מספר זה לא גדול.

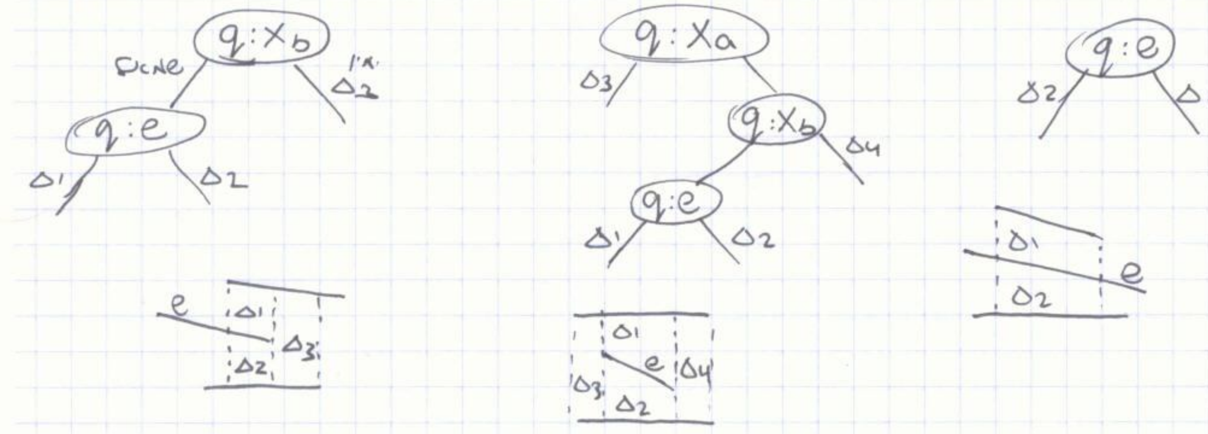
כאן גם, מספר הטרפזים שעלולים "לדרוך" על כל טרפז כתוצאה מהוספת צלע חסום ע"י 4 ולכן הוא $O(1)$ (קל לראות שזה 4). קל גם כך לבדוק בכל שלב לאיזה טרפז q "עוברת".
את הקשר בין הטרפזים נייצג ע"י גרף אציקלי מכוון (DAG) שצמתיו הם (טרפזים) השוואות משני סוגים:

השוואת x - האם q נמצאת משמאל/מימין לקואורדינטת ה x

השוואת y - האם q נמצאת מעל / מתחת לקטע

הסבר-

בגרף שלנו יהיו צמתים מהצורה הבאה



וביחד הם יצרו מעין עץ החלטה שעליו הם הטרפזים ב M

(אבל זה לאו דווקא עץ אם אנחנו מאחדים עלים שמייצגים את אותו הטרפז)

בכלל, נק' נכנסות כשאילתות למבנה הנתונים רק אחרי שבנינו אותו בשלמותו.

אבל, נקודות הקצה של הקטעים יכנסו למבנה במהלך הבנייה שלו וזה

כדי למצוא באיזה טרפזים נוכחיים הן נמצאות

הוספת קטע $ab=e$:

· נאתר את a במבנה הנוכחי ונמצא את הטרפז Δ_0 שמכיל אותה

· נתקדם לאורך e מ-Δ_0 ימינה ונמצא את $k+1$ הטרפזים ש-e חותך

משמאל לימין $\Delta_0, \Delta_1, \dots, \Delta_k$ $(O(k))$

· נעבור רגע נחליף טרפז נוכחי מעל e וטרפז נוכחי מתחת ל e.

כאשר כ"ו מהם מתחיל בקר שמאלי ידוע והאין פתוח וככה נגדיר

את הטרפזים החדשים.

ראינו איך האלגוריתם עובד. כעת נשאר לבדוק האם הוא יעיל. אני

נראה (1) תוחלת עומק ה DAG (זמן שאילתה) הוא $O(\log n)$

(2) תוחלת מס' הטרפזים שהאלג' מייצר היא $O(n)$

(3) תוחלת זמן ההכנה היא $O(n\log n)$

נתחיל מהוכחת (2)-

נסתכל על כל הטרפזים האפשריים, כלומר על כל הרביעיות הנקבעות ע"י

- קטע וקצה

- קטע תיקרה

- נק' קצה שנוצרת קיר שמאלי של אחד הקטעים האלה או של קטע חדש

- " " " ימני " " " " " " "

לכל טרפז Δ ניתן משקל w = מס' הקטעים שחוצים אותו

נשים לב שטרפז במשקל w יופיע במהלך הבניה אמ"מ 4 הקטעים המגדירים אותו יופיעו לפני ה w החוצים. ההסתברות לכך היא

$$\frac{4!\,w!}{(w+4)!} = 24/(w+1)(w+2)(w+3)(w+4) = O\left(\frac{1}{w^4}\right).$$

נסמן ב N_w את מס' הטרפזים במשקל W

ונסמן ב $N_{\le w}$ את מס' הטרפזים במשקל $\le W$

נקבל ש

$$\mathbb{E}(\#\text{טרפזים שנוצרו}) = \sum_{w\ge 0} \frac{24\, N_w}{(w+1)(w+2)(w+3)(w+4)} =$$

$$= O\left(\frac{N_{\le n-4}}{n^4} + \sum_{w=0}^{n-5} \frac{N_{\le w}}{w^5}\right) = \dots$$

עפ"י קלרקסון-שור $N_{\le w} = O(w^4 N_0(\frac{n}{w}))$.

קל לראות ש $N_0(\frac{n}{w})$ הוא מס' הטרפזים בפירוק הטרפזי של המפה של $\frac{n}{w}$ קטעים ולכן (עפ"י חישוב שראינו) $\le 3\cdot\frac{n}{w}+1 = O(\frac{n}{w})$.

נקבל (מה 2 האחרונים) $N_{\le w} = O(w^4\cdot\frac{n}{w}) = O(nw^3)$

$$\Rightarrow \dots = O\left(\sum_{w\ge 0}\frac{nw^3}{w^5}\right) = O(n)\cdot\sum_{w\ge 0}\frac{1}{w^2} = O(n)$$

הוכחת (1) -

נקבע נק' q קבועה ונחסום את תוחלת מס' הטרפזים שהאלגוריתם יוצר ושמכילים את q.

נסתכל על כל הטרפזים האפשריים שמכילים את q. נסמן סימונים כמו מקודם ואז ההסתברות להופעה של טרפז היא $O(\frac{1}{w^4})$ ותוחלת מספרם היא

$$O\left(\sum_{w} \frac{N^{(q)}_{\le w}}{w^5}\right)$$

(כאשר המשמעות של $N^{(q)}_{\le w}$ היא כמה טרפזים מכילים את q ומשקלם חסום ע"י W)

אנו מקבלים את אותו החישוב עפ"י Clarckson-Shur , רק שהפעם $N_0^{(q)}(r)=1$
ולכן $N_{\le\omega}^{(q)}=O(\omega^4)$ והתוחלת היא $O(n\log n) = O(\sum\frac{1}{\omega}) = O(\sum\frac{\omega^4}{\omega^5})$.

הוכחת (3) -
עלות הוספת קטע שווה לאיתור נקודת הקצה שמאלית + $O(k)$ (כאשר k הוא מס' הטרפזים שהקטע חצה). מרכיב העלות $O(k)$ מסוכם על כל הקטעים, והוא פרופורציוני למס' הטרפזים (מס' בתוחלת). תוחלת איתור נקודת הקצה היא $O(\log n)$ לקטע, וסה"כ $O(n\log n)$.

ניתוח הסיבוכיות בספר נעשה באופן שונה, עפ"י שיטה שנקראת Backwards-Analysis:
נסתכל על המפה לאחר הוספת k קטעים. קיימות הרבה אפשרויות לבחירת k קטעים אלו - נתמקד באחת מהן. אנו יודעים מי הם הקטעים, אך לא באיזה סדר הוספו. נסתכל על הקטע האחרון שהוספנו, e, שיכול להיות כל אחד מהקטעים הנ"ל בהסתברות $\frac{1}{k}$. אם נמחק את e, הטרפזים שיעלמו הם אלו ש-e משתתף בהגדרתם, ואלה בדיוק הקטעים הנוצרים בצעד האחרון. אם נסתכל על טרפז Δ המופיע במפה לאחר k הצעדים, ההסתברות שהוא נעלם בזמן מחיקת הקטע האחרון שהוכנס היא $\frac{4}{k}$. לכן תוחלת מספר הטרפזים שיעלמו עם מחיקת e היא
$\frac{4}{k}\times$ מס' הטרפזים $\sim 3k\cdot\frac{4}{k}=12$ $\Leftarrow$ חמו טרפזים בתוחלת.
כמובן שנדרשת הכרות עם המפה לאחר k צעדים