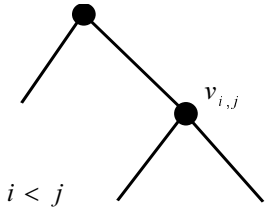
מטוסביץ' מרק
17 July, 1998

זמן ממוצע של Quick Sort

קלט: פרמוטציה של $n...1$.
למען האנליזה נניח כי איבר הציר לא מופיע לא במערך של קריאה לצד שמאל ולא לצד ימין.
בכל השוואה משתתף איבר ציר.
מסקנה: כל זוג i, j מושווה לכל היותר פעם אחת.
נסמן ב- $P(i, j)$ הסתברות להשוות בין i ל- j.

עץ הקריאות:
יהי v הצומת בעץ הקריאות של QS שמשותף ל- i ול- j.



טענה: כל $i < k < j$ גם כן מופיע ב- v.

$$P(i, j) \le \frac{2}{d}$$

טענה: יהי d מספר האיברים ב- v אזי
הוכחה:
i, j לא הרשוו באף אב הקדמון ממש של v (אחרת אחד מהם היה איבר הציר ואז לא היה מופיע ב- v)
אם i ו- j הושוו (ב- v) אזי או i או j היה איבר הציר. ההסתברות לכך היא $\frac{2}{d}$

$$d \ge j - i + 1 \quad \Rightarrow \quad P(i, j) \le \frac{2}{j - i + 1}$$

$$\sum_{i=1}^{n} \sum_{j>i} \left(P(i, j) \cdot 1 + \left(1 - P(i, j)\right) \cdot 0 \right) = \sum_{i=1}^{n} \sum_{j>i} \left(P(i, j) \right) = \sum_{i=1}^{n} \sum_{j>i} \frac{2}{j - i + 1}$$

תוחלת של מס' השוואות הכולל היא

כיון ש $1 \le i < j$ אזי $1 < j-i+1 \le j \le n$ ולכן כל איבר בסכום $\sum_{j>i} \frac{2}{j-i+1}$ מופיע בסכום $\sum_{j=1}^{n} \frac{2}{j}$.

לכן,

$$\sum_{i=1}^{n}\sum_{j>i}\frac{2}{j-i+1} < 2\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{j} = 2\sum_{i=1}^{n} H_n = 2nH_n = 2n\left(\ln n + O(1)\right) = 2n\ln n + O(n)$$

ומכאן זמן ממוצע של quick sort הוא $O(n\ln n)$.

עץ חיפוש בינארי אקראי

קלט: פרמוטציה של $1,...,n$

נכניס את המספרים לעץ חיפוש בינארי ריק בסדר אקראי.

התפלגות של עצי חיפוש בינאריים אקראיים זהה לזו של עצי הקריאות של Quick Search.

יהי d_i עומקו של צומת i:

$$\sum d_i = \mathrm{Time}(QS) = 2n\ln n + \mathrm{O}(n)$$

לכן העומק הממוצע שווה

$$\mathrm{E}(d) = \frac{1}{n}\left(2n\ln n + \mathrm{O}(n)\right) = 2\ln n + \mathrm{O}(1)$$

(הוכחה שהגובה הצפוי שווה $\mathrm{E}(h(\mathrm{T})) = \mathrm{O}(\ln n)$ מופיעה ב- Cormen)

Treap (Aragon and Seidel)

קלט: מפתחות $x_1,...,x_n$

נגריל קדימויות $p_1,...,p_n$ בהסתברות אחידה בקטע $[0,1]$.

Treap – עץ חיפוש בינארי על $x_1,...,x$ אבל ה- p_i -ים יקיימו את סדר הערימה: p_i של האבא קטן מ- p_j של הבנים. בשורש תהיה רשומה i עבורה p_i מינימלי.

טענה: בהינתן $x_1,...,x$ (שונים) ו- $p_1,...,p$ (שונים) Treap נקבע באופן חד ערכי.

אלגוריתם לבניית Treap:

בחר את p_i המינימלי כשורש, הפרד ל- $x_1^l,...,x_{n_l}^l < x_i$ ול- $x_1^r,...,x_{n_r}^r > x_i$ והמשך בכל תת פרמוטציה באופן רקורסיבי.

טענה: התפלגות ה- Treap זהה להתפלגות עצי חיפוש בינאריים על n צמתים לכל $x_1,...,x$ (שונים).

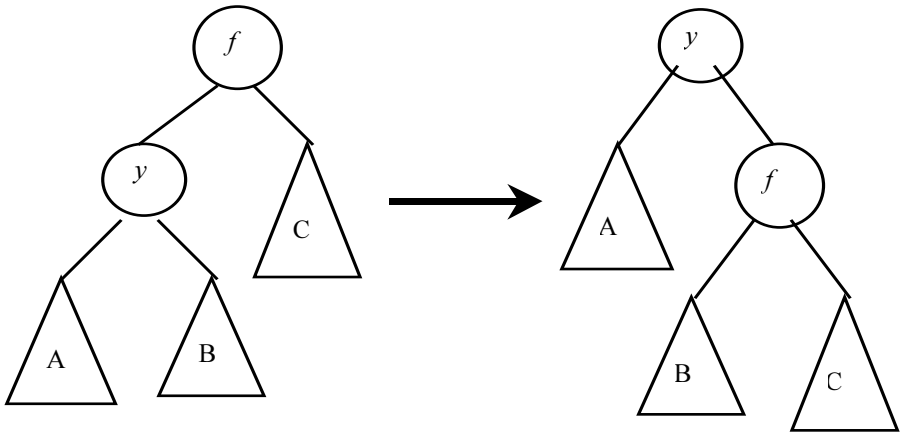
הסיבה לכך היא שאם נסדר את $p_1,...,p_n$ כך ש-

$$p_{\pi(1)} < p_{\pi(2)} < ... < p_{\pi(n)}$$

ונכניס את $x_1,...,x_n$ לפי סדר π, נקבל בדיוק את ה- Treap שיתקבל מ- $\bar{x}$ ו- $\bar{p}$.

פעולות על Treap:

1) חיפוש: חיפוש על עץ חיפוש בינארי רגיל (מתעלמים מ- p);
2) הכנסת y:
   א) נגריל p;
   ב) אם f הוא אבא של y ו- $p_f > p_y$ נבצע גלגול:



נשים לב כי אחרי גלגול תת העץ של f היינו חוקי. נמשיך בתהליך עד אשר נגיע לשורש או $p_{\text{father}} < p_y$

3) הוצאת את y:
להגדיל p לאינסוף, לבצע גלגולים ואז להוציא כעלה.

Backward Analysis

דוגמא:

בהינתן רשימה מקושרת באורך n המסודרת בסדר שרירותי, ממיינים אותה בעזרת אלגוריתם הבא:

1) בוחרים באקראי איבר x ומחלקים את הרשימה לשני חלקים: בחלק אחד כל איברים שקטנים מ- x ובחלק שני כל אלה שגדולים ממנו (את ה- x עצמו שמים במערך שיכיל את תוצאת המיון (מיקומו כבר ידוע));
2) מתוך הקטעים שהתקבלו בוחרים קטע באקראי, ומתוך קטע זה בוחרים איבר באקראי. כעת מחלקים את הקטע לשני חלקים (כמו קודם) ורושמים את האיבר שבחרנו למערך וממשיכים כך כל עוד יש קטעים.

מהי סיבוכיות ממוצע של האלגוריתם?

כדי לפצל קטע נצטרך זמן כאורך הקטע.
נסמן ב- S_i את קבוצת ה- pivots שנבחרו עד השלב ה- i (כולל).

בשלב ה- i יש $i+1$ קטעים שמכילים יחד $n-i$ איברים. לכן גודל קטע ממוצע היינו $\frac{n-i}{i+1}=\mathrm{O}\left(\frac{n}{i}\right)$.

בתהליך בונים את S_i על סמך S_{i-1} והתלות מסובכת (אורך הקטעים תלוי בבחירות קודמות). אבל עם נסתכל על התהליך מתחילתו עד סיומו אז החישוב יהיה קל יותר: בהינתן S_i מהי ההסתברות שהיא התקבלה מ- S_{i-1}. כל אחד מאיברי S_i בהסתברות שווה הגיע אחרון ולכן תוחלת גודל הקטע שחילקנו תוך שלב i היא

$$2\mathrm{E}(k)=2\,\mathrm{O}\left(\frac{n}{i}\right)=\mathrm{O}\left(\frac{n}{i}\right)$$

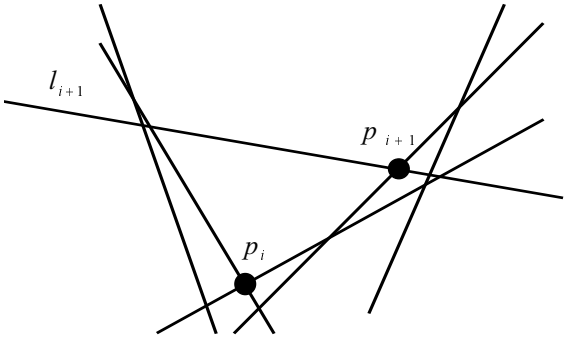
כאן סימנו ב- $\mathrm{E}(k)$ תוחלת של גודל קטע המתקבל אחרי החילוק.
סך הכל האלגוריתם דורש זמן

$$\sum_{i=1}^{n}\mathrm{O}\left(\frac{n}{i}\right)=\mathrm{O}\left(n\sum_{i=1}^{n}\frac{1}{i}\right)=\mathrm{O}(n\ln n)$$

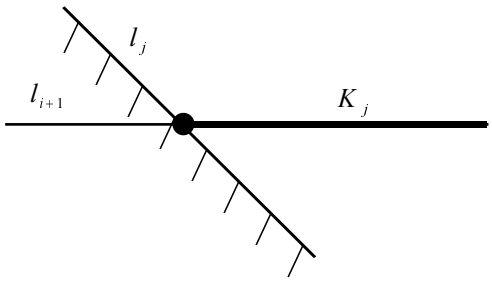
2- Dimensional Linear Programming

בעיה:
נתונים n ישרים במישור. צריך למצוא נקודה בעלת קואורדינטה y מינימלית כך שהנקודה נמצאת מעל כל הישרים.

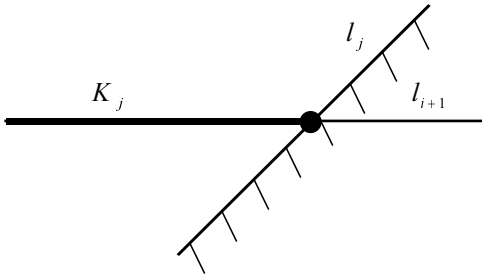
אלגוריתם:
סדר את הישרים בסדר אקראי $l_1,...,l_n$.
תהי p_2 נקודת החיתוך של l_1,l_2 (נניח כי הם בעלי שיפועים הפוכים בסימן).
בשלב ה- i נשמור נקודה p_i שהיא המינימום עבור הישרים $l_1,...,l_i$.
במעבר משלב i לשלב $i+1$: אם l_{i+1} נמצא מתחת ל- p_i אז $p_{i+1}=p_i$, אחרת p_{i+1} נמצאת על הישר l_{i+1} ואז נצטרך למצוא על l_{i+1} את הנקודה בעלת קואורדינטה y מינימלית הנמצאת מעל כל הישרים $l_1,...,l_i$.



לשם כך נסתכל על l_{i+1}. חיתוך חצי המישור המוגדר ע"י l_j קובע על l_{i+1} קרן K_j אין סופית של נקודות הנמצאות מעל הישר l_j.



או



לכן p_{i+1} היא הנקודה בעלת y מינימלי מבין $\left(\bigcap_{j\le i} K_j\right)\cap l_{i+1}$. אבל $\left(\bigcap_{j\le i} K_j\right)\cap l_{i+1}$ הוא קטע ולכן הנקודה p_{i+1} היא אחד מקצוותיו. וקצוות הקטע אפשר למצוא בקלות תוך i צעדים (בעצם זאת אותה הבעיה אבל חד ממדית).

סיבוכיות:
ההסתברות לביצוע תיקון בשלב i שווה להסתברות שהישר l_i נוגע בנקודה p_i (נקודת המינימום עבור הישרים $(l_1,...,l_i)$, כלומר ההסתברות היא $\frac{2}{i}$. זמן ביצוע תיקון הוא $\mathrm{O}(i)$, לכן הזמן הצפוי של צעד i הוא

$$\mathrm{P}(\text{יש תיקון})\cdot \mathrm{O}(i) + \mathrm{P}(\text{אין תיקון})\cdot 1 = \frac{2}{i}\cdot i + \mathrm{P}(\text{אין תיקון}) = \mathrm{O}(1)$$

סה"כ תוחלת זמן ריצת האלגוריתם היא

$$\sum_{i=1}^{n} \mathrm{O}(1) = \mathrm{O}(n)$$