מבני נתונים ואלגוריתמים - הרצאה 9

29 בנובמבר 2011

סיכום מיונים

חילקנו מיונים למיוני השוואה ומיונים שהם לא השוואתיים.
במיוני השוואה ראינו:

- Bubble Sort
- Selection Sort
- Insertion Sort
- Cocktail Sort

שהם כולם $O\left(n^2\right)$.

- Quick Sort

עלות טיפוסית של $O\left(n\log n\right)$ אבל במקרה הכי גרוע $O\left(n^2\right)$

- Heap Sort - **לא יציב**
- Merge Sort

עלות של $\Theta\left(n\log n\right)$.
ראינו משפט שאמר שכל מיון השוואה הוא $\Omega\left(n\log n\right)$.
במיונים שאינם מיוני השוואה ראינו:

- Counting Sort - לספור כמה פעמים כל איבר מופיע. עלות של $O\left(n+k\right)$ כאשר מס' האיברים השונים שיש.
- Pidgeon Hole Sort - מחזיקים מקום לכל ערך אפשרי ומכניסים כל איבר למקום שלו.
- Bucket Sort
- Radix Sort - חילקנו לשני סוגים:
  Least Significant Digit - לפי סדר ערכים, אלגוריתם איטרטיבי.
  Most Significant Digit - לפי סדר לקסיקוגרפי, אלגוריתם רקורסיבי.
  הסיבוכיות של המיון היא $O\left(n\cdot L\right)$ כאשר n מס' הערכים וL אורך הייצוג בביטים.
  אם כל הערכים שונים, אז יש לכל היותר 2^L מספרים אפשריים, כלומר $n \leq 2^L$ ואז $L\geq\log_2 n$.
  אם יש הרבה חזרות, אז יתקיים $n\geq 2^L$ ואז $L\leq\log n$ ואז נקבל שRadix Sort יעיל.

גרפים

גרפים מתחלקים לגרפים מכוונים וגרפים לא מכוונים.
אם לקשתות יש כיוון, כלומר היא הולכת מקדקד אחד לקדקד אחר, הגרף מכוון.
אם הקשתות רק מחברות בין קדקדים בלי חשיבות לסדר אז הגרף לא מכוון.
נסמן גרף כ$G=\left(V,E\right)$ כאשר V קב' הקדקדים וE קב' הקשתות.
קשת e_i היא זוג של קדקדים - בגרף מכוון הזוג סדור ובגרף לא מכוון הזוג לא סדור.
גרף פשוט הינו גרף שבו אין קשתות עצמיות (קשת מקדקד לעצמו) ואין ריבוי קשתות (אין קשתות כפולות).
קשירות: נאמר שא' קשיר לב' אם קיים הילוך בין א' לב'.
קשירות היא יחס שקילות בגרף לא מכוון.
קשת חותכת היא קשת שהסרה שלה מהגרף תגדיל את מס' רכיבי הקשירות.

טענה

קשת היא חותכת $\iff$ היא לא חלק ממעגל.

הוכחה

נניח e קשת חותכת וחלק ממעגל. קיימים x וy שקשירים בG ולא קשירים ב$G\setminus\{e\}$.
לכן, e חלק מהמסלול בין x לy.
$e=(u,v)$ חלק ממעגל. קיים מסלול מu לv שלא דרך e שסוגר מעגל, נסמנו בL.
$L\subseteq G\setminus\{e\}$, לכן קיים מסלול בין x לy בו נחליף את e בL שמוכל ב$G\setminus\{e\}$, מש"ל צד א'.
נניח שe לא חותכת ולא חלק משום מעגל.
e מחברת בין u לv בגרף G. u, v לא מחוברים בהילוך ב$G\setminus\{e\}$ כי אחרת ניתן היה לסגור מעגל, מש"ל ב'.

מסקנה

כל קשת בעץ היא חותכת.

הגדרה

עץ G פורש את גרף G אם כל הקדקדים של G קשירים בT.

הגדרה

עץ הוא מינימלי אם סכום משקלי הקשתות הוא מינימלי - Minimal Spanning Tree, MST.

מציאת עץ פורש מינימלי

אלגוריתם 1 מציאת עץ פורש מינימלי - Reverse Delete

1. סדר את הקשתות מן היקרה ביותר לזולה ביותר.
2. עבור על כל קשת. אם קשת לא חותכת, הורד אותה.
3. סיים כאשר מס' הקשתות הוא מס' הקדקדים פחות 1.

אך צריך אלגוריתם אחר, לבדוק האם קשת היא חותכת:

אלגוריתם 2 בדיקה האם קשת חותכת

1. בצע BFS וספור רכיבי קשירות.
2. הורד קשת.
3. בצע שוב BFS וספור אם יש הבדל.

אלגוריתם נוסף למציאת עץ פורש מינימלי:

אלגוריתם 3 אלגוריתם Kruskal למציאת MST

1. סדר את הקשתות מן הזולה ביותר ליקרה ביותר.
2. עבור על כל קשת. אם היא מחברת שני רכיבי קשירות בעץ, הוסף אותה לעץ.
3. סיים כאשר הגעת למס' קשתות כמס' הקדקדים פחות 1.

אלגוריתם 4 אלגוריתם Prim למציאת MST

1. בחר קדקד באקראי והכנס לT.
2. הכנס את שאר הקדקדים לS.
3. מצא קשת זולה ביותר בין S לT. הוסף אותה לעץ, ואת הקדקד שמחובר אליה בS הוסף לT.
4. סיים כאשר S ריק.

יעילות האלגוריתמים

אלגוריתם BFS עולה $O\left(E\right)$.
באלגוריתם Kruskal וReverse Delete אנו עוברים על כל קשת ועושים BFS ולכן זה $O\left(E^2\right)$. המיון עצמו הוא $O\left(E\log E\right)$ וזה זניח לעומת $O\left(E^2\right)$ לכן סה"כ זה $O\left(E^2\right)$.
עבור אלגוריתם Prim:
המיון הוא $O\left(E\log E\right)$.
אנו עוברים על כל הקדקדים ולכל קדקד עוברים מקסימום על כל הקשתות לכן זה בערך $O\left(V\cdot E\right)$.
בהכרח מס' הקדקדים קטן ממס' הקשתות כי אחרת הגרף לא קשיר ואין עץ פורש, לכן Prim יעיל יותר מהשניים האחרים.