אלגוריתמים

פיתרון תרגיל בית 3

1. האלגוריתם: נריץ את האלגוריתם של קרוסקל על קשתות העץ T ביחד עם הקשתות החדשות שנוספו. בגלל ש $|E| \leq 2|V|$ הסיבוכיות היא $O(VlogV)$.

2. נחשב עץ פורש מינימלי T, את כל קשתותיו נצבע בכחול ואת שאר הקשתות באדום. נריץ את האלגוריתם למציאת עפ"מ עם מספר מקסימלי של קשתות אדומות מהתרגול. אם קיבלנו עץ עם קשת אדומה, נחזיר "כן", ואחרת יש לגרף עפ"מ יחיד, ולכן נחזיר "לא".

3. האלגוריתם: נמיין בזמן ליניארי את הקשתות (נשמור מערך בגודל $|V|$ של רשימות, בכל כניסה i יהיו כל הקשתות במשקל i). נריץ את האלגוריתם של קרוסקל, כאשר בכל פעם שמסיימים לעבור על רשימה של קשתות במשקל i, נעבור על רשימת הקשתות במשקל $i+1$ ועבור כל אחת נבדוק האם שני צדדיה באותו רכיב קשירות (בעזרת שתי פעולות $find$), ואם כן נחזיר אותה (ואחרת לא). נשים לב שלא שינינו את אלגוריתם קרוסקל, רק הוספנו פעולות שלא משפיעות על האלגוריתם.

4. נגדיר את התתי בעיות - $c[k,j]$להיות המחיר המינימלי של j חלוקה של המספרים $a_1 \ldots a_k$. כלומר מחפשים את $c[n,d]$. בנוסף, $c[1,1]=a_1$, ומתקיים:
$$c[k,j] = min_{i<k}\{max\{c[i,j-1], \sum_{f=i+1}^{k} a_f\}$$
כלומר פתרון אופטימלי עבור $c[k,j]$ ניתן לבנות מפתרון אופטימלי עבור המספרים $a_1 \ldots a_i$ עבור $i<k$ כלשהו וחלוקה בגודל $j-1$, ושאר האיברים $a_{i+1} \ldots a_k$ נמצאים בקבוצה האחרונה בחלוקה.
סיבוכיות: nd תתי בעיות, כל אחת לוקחת במקרה הגרוע $O(n)$, סה"כ $O(n^2d)$.

5. נסתכל על המספרים כעל מערך $a_1 \ldots a_n$. נגדיר את תתי הבעיות $c[i,j,k]$ להיות ערך המשחק עבור התת־מערך $a_i \ldots a_j$ כאשר השחקן k הוא הבא בתור לשחק. כלומר מחפשים את $c[1,n,A]$. כמובן ש־ $c[i,i,A]=a_i$ וגם $c[i,i,B]=-a_i$. לפי הגדרת השחקנים:
$$c[i,j,A] = max\{c[i+1,j,B]+a_i, c[i,j-1,B]+a_j\}$$
$$c[i,j,B] = min\{c[i+1,j,A]-a_i, c[i,j-1,A]-a_j\}$$

6. נמיין את המגשים לפי השטח, מהגדול לקטן (לוקח $O(n \cdot \log n)$). נגדיר את תתי הבעיות $c[k]$ להיות הערימה הגדולה ביותר של מגשים כאשר k המגש העליון. כלומר מחפשים את $max_k\{c[k]\}$. כמובן ש־ $c[1]=1$. נגדיר לכל k את קבוצת המגשים $M[k]$ שניתן להניח עליהם את k, אז $c[k] = max_{j \in M[k]} c[j]+1$. עבור כל k לוקח $O(n)$ לחשב את $M[k]$ במקרה הגרוע, ולכן סה"כ $O(n^2)$ לחשב את כל M. עבור כל k לוקח $O(n)$ לחשב את $c[k]$ במקרה הגרוע, ולכן גם בסה"כ $O(n^2)$.

7. נגדיר את $f(k,j)$ להיות הערך המקסימלי שניתן להשיג עבור הסכום $\sum_{i=1}^{j} f_i(x_i)$ תחת האילוץ $x_1+x_2+\ldots+x_j=k$. נגדיר את $d(k,j)$ להיות וקטור הערכים $(x_1,\ldots,x_j)$ המהווים פתרון אופטימלי עבור $f(k,j)$. כלומר מחפשים את $d(t,m)$ עבור t שמקיים $f(t,m) = max_k f(k,m)$. כמובן ש־ $f(k,1)=f_1(k)$, וגם $d(k,1)=(k)$. פתרון אופטימלי עבור $f(k,j)$ מורכב מערך כלשהו עבור x_j, ושאר הערכים ממקסמים ביחס לסכום שלהם, כלומר $f(k,j) = max_{t=1}^{k}\{f(t,j-1)+f_j(k-t)\}$, בנוסף $d(k,j) = (d(t,j-1),k-t)$ עבור הערך t שמביא למקסימום. יש m^2 תתי בעיות, כאשר כל בעיה ניתנת לחישוב במקרה הגרוע ב $O(m)$. לכן סה"כ הסיבוכיות $O(m^3)$.

8. נריץ k איטרציות של בלמן פורד. זמן הריצה הוא $O(k|E|)$. כיוון שבסוף כל איטרציה j אנו מחזיקים את המרחקים של כל הצמתים שהמסלול הקל ביותר אליהם הוא באורך j קשתות, לאחר k איטרציות לפי נתון השאלה נחזיק את המסלולים הקצרים ביותר עבור כל הצמתים.