מבני נתונים ואלגוריתמים - הרצאה 11

4 בדצמבר 2011

מסלולים מינימליים

- אם כל הקשתות במשקל שווה, פשוט מפעילים BFS.
- קשתות שונות אבל חיוביות - Dijkstra:
מחלקים לשתי קבוצות קדקדים. מתחילים כשבS נמצא רק v_i וב$V \backslash S$ את השאר.
מגדירים

$$d_j = \begin{cases} v_{i,\,j} & i,\, j \text{ are neighbours} \\ \infty & \text{else} \end{cases}$$

רלקסציה:

$$d_{ij} = d_j = \min\left(d_j,\, d_k + w_{kj}\right)$$

כאשר $k \in S$.
בוחרים קדקד עבור d_j מינימלי ב$V \backslash S$ ומעבירים אותו לS.
באופן מעשי, מחשבים את d_{ij} רק עבור שכנים של קדקד שהוכנס לS בשלב קודם.
עלות כוללת:
העדכון עולה $\frac{E}{V}$ = מס' שכנים ממוצע של כל קדקד.
בחירת המינימום עולה V.
העדכון ובחירת המינימום קורה בכל שלב, ויש לנו שלב לכל קדקד, לכן סה"כ העלות היא $O\left(V \cdot \left(\frac{E}{V} + V\right)\right) = O\left(E + V^2\right) = O\left(V^2\right)$.
אם משתמשים בערימה, העלות יורדת בהרבה כי אז בחירת המינימום לוקחת $\log V$ ואז העלות הכוללת היא $O\left(E + V \log V\right)$.

- אם הקשתות לא בהכרח חיוביות, משתמשים בBellman Ford - כמו Dijkstra אבל מבצעים רל-קסציה על כל הקדקדים.
העלות היא $O\left(V \cdot E\right)$.
באלגוריתם זה, יש להניח שהגרף מכוון.

אלגוריתם Floyd-Warshall

אלגוריתם שמוצא את כל המסלולים הקצרים ביותר בגרף.
ניתן לכל קדקד מספר סידורי, ונגדיר d_{ij}^k כמרחק המינימלי בין i לj עם הקדקדים הפנימיים שמספרם לכל היותר k.

$$d_{ij}^0 = \begin{cases} w_{ij} & i,\, j \text{ are neighbours} \\ \infty & \text{else} \end{cases}$$

יכול להיות מצב שבו להוסיף את k לא באמת עוזר, כלומר יש מסלול לא יותר יקר שמכיל מס' עד $k-1$ מאשר מסלול שיכול להכיל קדקדים עד k, ואז

$$d_{ij}^k = d_{ij}^{k-1}$$

ואם k כן נמצא במסלול הפנימי, אז מתקיים

$$d_{ij}^k = d_{ik}^{k-1} + d_{kj}^{k-1}$$

לכן

$$d_{ij}^k = \min\left(d_{ij}^{k-1},\, d_{ik}^{k-1} + d_{kj}^{k-1}\right)$$

האלגוריתם הוא:

אלגוריתם 1 אלגוריתם Floyd-Warshall למציאת כל המסלולים הקצרים בגרף

נגדיר $d_{ij}^{0}=w_{ij}$ אם $i,\,j$ שכנים ו-∞ אחרת.

for $(k=1;\ k\leq|V|;\ k++)$

$d_{ij}^{k}=\min\left(d_{ij}^{k-1},\,d_{ik}^{k-1}+d_{kj}^{k-1}\right)$

עלות הריצה היא $O\left(V^{3}\right)$ ועלות הזיכרון היא $O\left(V^{2}\right)$.

ייצוג של גרף

1. ייצוג ע"י רשימת קשתות - שלשות של מספרים $(i,\,j,\,w)$ מסמנת קשת מ-i ל-j עם משקל w.
מחזיקים את הרשימה ממוינת לפי מקור ואז מטרה.
חיפוש קשת תיקח $O\left(\log E\right)$, ועלות הזיכרון היא $O\left(E\right)$.

2. ייצוג על ידי מטריצת שכנויות:

$$A_{ij}=\begin{cases} w_{ij} & i,\,j\text{ are neighbours}\\ -\infty & \text{else} \end{cases}$$

ואז חיפוש קשת תיקח $O\left(1\right)$ אבל עלות הזיכרון היא $O\left(V^{2}\right)$.

3. טבלת גיבוב (Hash Table) - נלמד בהמשך.

4. מערך של מצביעים לרשימות. כל תא במערך מייצג קדקד והרשימה שיוצאת ממנו היא רשימת הקדקדים אליהם יש לו קשת.
מציאת קשת במקרה זה היא $O\left(\frac{E}{V}\right)$ (מס' השכנים הממוצע).
אם נחזיק את הקשתות בעץ חיפוש ולא ברשימה, נקבל שמציאת קשת תיקח $O\left(\log\left(\frac{E}{V}\right)\right)$.

אנטרופיה - אי ודאות

התפלגויות מתחלקות לשני סוגים:
רציפה - $P\left(x\right)$ או בדידה - P_{i}.
בהתפלגות בדידה, נגדיר משתנה אנטרופיה:

$$H=-\sum_{i}P_{i}\log_{A}\left(P_{i}\right)$$

במקרה הקיצוני שבו ההתפלגות היא כך ש-$P_{1}=1$ ולכל $i\neq1$ $P_{i}=0$ נקבל:

$$H=0$$

ואז כמות אי הוודאות במערכת היא 0, מינימלית.
במקרה הקיצוני ההפוך, שבו $P_{i}=\frac{1}{k}$ לכל $i\in\{1,\,...,\,k\}$, אזי

$$H=-\sum_{i=1}^{k}\frac{1}{k}\log_{A}\left(\frac{1}{k}\right)=-k\cdot\frac{1}{k}\log_{A}\left(\frac{1}{k}\right)=-\log_{A}\left(\frac{1}{k}\right)=\log_{A}\left(k\right)$$

אם נבחר $A=k$ אז נקבל $H=1$.
כך, נקבל סולם אי ודאות בין 0 ל-1, כאשר $H=0$ אומר שאין אי ודאות וכאשר $H=1$ יש חוסר ודאות מוחלט.

קוד רישא

קוד רישא הוא קידוד של אותיות כך ששום אות היא לא רישא של אות אחרת.
כלומר לא יכול להיות שלמשל נקדד את א' להיות 01011 ואז ב' יהיה 010111 (כי אז א' רישא של ב').
נגדיר לכל אות\מילה אורך ℓ_i - אורך המילה בייצוג של המחשב. אזי אורך המילה הטיפוסי יהיה התוחלת:

$$L = \sum_i P_i \ell_i$$

רוצים שL יהיה קטן.

טענה

בקוד רישא מתקיים:

$$L \geq k \cdot H$$

(k מספר האותיות, H האנטרופיה).

קוד הופמן

קוד רישא ניתן להציג באמצעות עץ, שבו כל אות\מילה נמצאת בעלה, כל פיצול שמאלה הוא 0 וכל פיצול ימינה הוא 1.

1. חבר את שתי האותיות\מילים עם שכיחות מינימלית, החשב אותן לאות אחת שהשכיחות שלה היא חיבור שתי השכיחויות.
2. חזור לא' כל עוד יש איברים לא מחוברים.

בצורה זו, נקבל עץ שבו האות\מילה עם השכיחות הכי גבוהה נמצאת הכי למעלה (כלומר היא הכי קצרה מבחינת ייצוג בביטים) ואילו המילה\אות עם השכיחות הכי נמוכה נמצאת הכי למטה (כלומר הכי ארוכה מבחינת ייצוג בביטים).