12/11/09

אלגוריתמים ברשתות -

- מימוש אלג' push-relabel

- השגת זמן ריצה $O(n^3)$

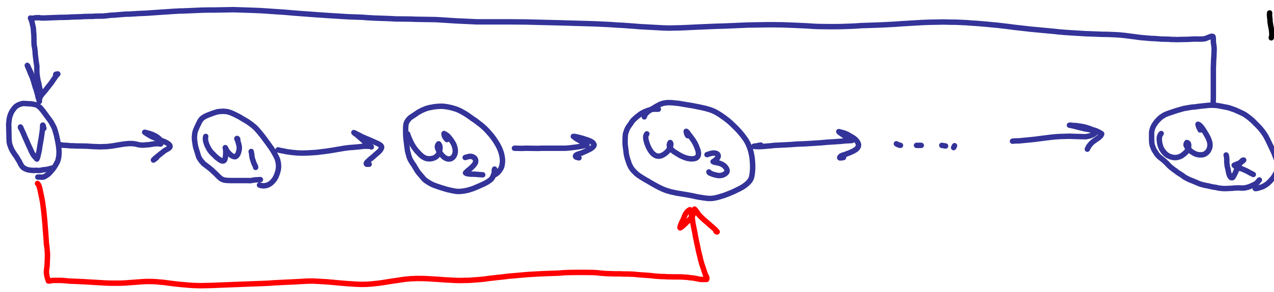
מימוש אלגוריתם push-relabel

גרף תשתית: יהי $G=(V,E)$ גרף מכוון. גרף התשתית של G הוא גרף לא מכוון $\tilde{G}=(V,\tilde{E})$ שבו $w \to v \in E$ או $v \to w \in E$ $\iff$ $\{v,w\} \in \tilde{E}$

נקצר ונאמר $\{v,w\}$ קשת של מכוונת ב G.

לכל קשת לא מכוונת $\{v,w\}$ נחזיק 3 ערכים: $c(v,w)$, $c(w,v)$, $f(v,w)$ [שווה ל $-f(w,v)$].

לכל צומת v נחזיק רשימת הקשתות הלא מכוונות שמכילות אותו



קשת נוכחית (מועמדת לדחיפה)

אם v פעיל, בודק האם ניתן לדחוף כמה לאורך הקשת הנוכחית. אחרת, מקדם את הקשת הנוכחית.

אם מגיע לסוף הרשימה, מעדכן תווית של v, והקשת הנוכחית מצביעה לשכן הראשון w_1.

נדון בתחזוקת קבוצת הצמתים הפעילים בהמשך.

Push/Relabel(v) [בטיפול בצומת v]

1) תהי $\{v,w\}$ הקשת הנוכחית ברשימה של v.

2) בצע push(v,w) אם ניתן. (אל תבצע שלבים 4,3)
אחרת

3) אם w אינו השכן האחרון ברשימה של v,
קדם את המצביע של הקשת הנוכחית לשכן הבא.

4) אם w אחרון: עדכן את המצביע של הקשת
הנוכחית לשכן הראשון, ובצע relabel(v).

טענה 4.1: בשורה (4) התנאים לביצוע relabel(v) מתקיימים.

הוכחה: רשימת שכני v נסרקה בטרם הגענו לשורה 4.
בכל קידום של המצביע לקשת הנוכחית מ-w_i ל-w_{i+1} מובטח
שמתקיים: $r_f(v,w_i)=0$ או $d(v) \le d(w_i)$.
אם relabel(v) לא אפשרי, משמע קיים שכן w_i ש"מפריע".
כלומר: $r_f(v,w_i) > 0$ וגם $d(v) > d(w_i)$.
אם $r_f(v,w_i)=0$ בזמן הקידום של המצביע ועבור $r_f(v,w_i)>0$
בהגעה לשורה 4, משמע שבוצע ביניהם push(w_i,v).
בזמן ה-push מתקיים $d(w_i) > d(v)$. ולכן בשורה (4) עדיין
$d(v) < d(w_i)$.
אם $d(v) \le d(w_i)$ בזמן הקידום של המצביע, אזי $d(v)$ נשאר
כמותו עד בשורה (4) כיוון ש-$d(w_i)$ לא ירד (ואולי עלה).
מסקנה: w_i אינו יכול להפריע לביצוע relabel(v). $\blacksquare$

תחזוקת קבוצת הצמתים הפעילים

v פעיל יכול להפסיק
להיות פעיל רק ע"י
push/relabel(v)

Q = קב' הפעילים

בתחילה $Q = \{v \in V \setminus \{s,t\} : c(s,v) > 0\}$

בכל פעולת $push(v,w)$: $insert(Q,w)$ (יתכן ש-w כבר פעיל).

ואם הדחיפה לא מרווה אז $delete(Q,v)$.

בכל פעולת $push/relabel(v)$ השינוי ב-Q ממומש ב $O(1)$.

רשימה מקושרת דו-כיוונית של Q

$$first(Q) \rightarrow \boxed{1} \rightleftarrows \boxed{2} \rightleftarrows \boxed{3} \rightleftarrows \cdots \rightleftarrows \boxed{K} \leftarrow last(Q)$$

לכל צומת v: $active(v) \in \{0,1\}$ מציין האם $v \in Q$

$Q\_ptr(v)$ מצביע למקום של v ב-Q.

האלגוריתם push/relabel

אתחל: קדם פרימה f, פונקצית תוויות d, קד׳ פעילים Q

כל עוד $Q \neq \emptyset$ עשה

יהי $v \in Q$ צומת פעיל
בצע push/relabel(v)
עדכן את Q

החזר f.

טענה 4.2: זמן הריצה של אלגוריתם push/relabel הוא

$$O(n \cdot m) + C \cdot \#\text{nonsaturating pushes}$$

$$= O(n^2 m)$$

הוכחה: נתמקד בצומת v, ונשאל: כמה פעמים נסרקת הרשימה של v? לשים לב שכל סריקה (למעט האחרונה) מסתיימת ב-$relabel(v)$, ולכן מס' הסריקות קטן מ-$2n$.

אורך הרשימה הוא $deg(v)$, ולכן סה"כ מס' הקידומים של המצביע בקשת הנוכחית (עבור כל הצמתים) חסום ע"י

$$O\left(n \sum_v deg(v)\right) = O(n \cdot m).$$

רק בדחיפה של מקודם המצביע.

מס' הדחיפות המרוות $= O(nm)$ (נבלע בחישוב למעלה).

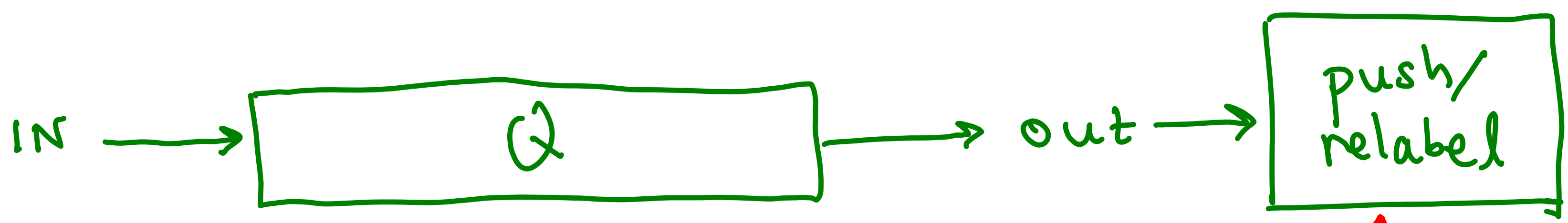
מס' הדחיפות הלא מרוות $= O(n^2 m)$.

כל הפעולה של push/relabel לוקחת $O(1)$, מכאן משל. □

שיפור זמן הריצה של האלגוריתם

הרעיון המרכזי: קביעת מדיניות ניהול התור

ניתן להשיג סיבוכיות $O(n^3)$ במספר שיטות.
הטבעית והפשוטה היא FIFO (first-in first-out).



מעבר על הפעולה על
v עד אשר v לא פעיל
או d(v) גדל

Discharge(Q)

1) $v \leftarrow$ dequeue(Q)

2) repeat

3) push/relabel(v)

4) if w becomes active, then insert(Q, w)

5) until $e(v) = 0$ or $d(v)$ increases.

6) if $e(v) > 0$, then insert(Q, v)

השגרה מופעלת כל
עוד $Q \neq \emptyset$.
שמורה: כל אברי Q הם
צמתים פעילים.

הערה: ניתן להחליף שורה (5) ב- until $e(v)=0$

ניתוח הזמן כאשר Q מנוהל בשיטת FIFO

$Q_0=\{v\in V\setminus\{s,t\}: c(s,v)>0\}$

PASS 1: סדרת פעולות discharge שמבוצעות כל עוד התור מחזיק ברציפות צמתים שהיו בתור מאז האיתחול.

Q_1: קב' הפעילים בסוף PASS 1.

$v\in Q_1 \Leftrightarrow$ הוכנס לתור במהלך PASS-1.

בהינתן Q_i, נגדיר PASS$(i+1)$ בתור סדרת פעולות discharge שמבוצעות החל מסוף PASS i כל עוד התור מכיל ברציפות צמתים מ Q_i.

טענה: מספר המעברים ע"פ התור $\geq 4n^2$.

הוכחה: נגדיר $\Phi_i \triangleq \max\{d(v) : v \in Q_i\}$.

כיצד מושפעת Φ ממעבר בודד על התור $\left(\Phi_i \lesseqgtr \Phi_{i+1} \right)$?

אם לא בוצעו פעולות relabel, הרי שצמתים הנמצאים בכל

צמתי פעיל ב־Q_i הועברו לצמתים "נמוכים" יותר, ולכן

התוויות של כל צמתי Q_{i+1} קטנים מ־Φ_i.

$\Leftarrow$ ירידה בערך של Φ. $\left(\Phi_{i+1} < \Phi_i \right)$.

ולכן אם $\Phi_{i+1} \geq \Phi_i$, משמע בוצע relabel במהלך PASS(i+1).

מס' פעולות relabel חסום ע"י $2n^2$, ולכן מס' מעברים שבהם

$\Phi_{i+1} \geq \Phi_i$ גם חסום ע"י $2n^2$.

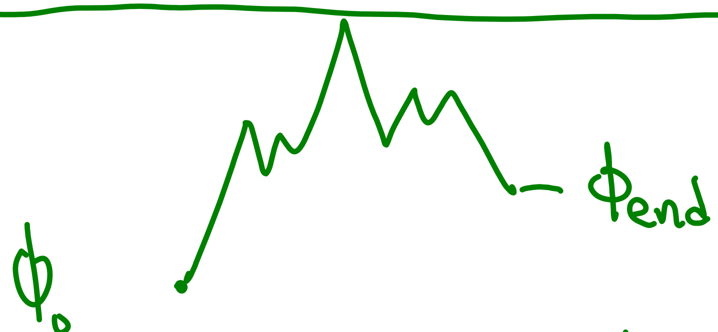
כלומר נחסום את $\left|\{i \mid \phi_{i+1} < \phi_i\}\right|$.

$$\Delta^+ \triangleq \sum_{\{i : \phi_{i+1} \geq \phi_i\}} \left(\phi_{i+1} - \phi_i\right) \leq \sum_{\substack{\text{פעולות} \\ \text{relabel}}} \underbrace{\left(d_{new}(v) - d_{old}(v)\right)}_{\text{העלייה בגובה של } v \text{ כתוצאה מפעולת } relabel(v)}$$

$$\leq \sum_v \left(d_{max}(v) - d_0(v)\right) \leq n \cdot 2n$$

מאידך,

$$\left|\{i : \phi_{i+1} < \phi_i\}\right| \leq \underbrace{\sum_{\{i : \phi_{i+1} < \phi_i\}} \left(\phi_i - \phi_{i+1}\right)}_{\Delta^-}$$



$\phi_{end} - \phi_0 = \Delta^+ - \Delta^-$

$$= \Delta^+ + \phi_0 - \phi_{end} \leq 2n^2$$

מהו זמן הריצה הכרוך ב-$4n^2$ מעברים ע"פ התור?

בכל הוצאה מהתור מבוצע push/relabel(v). נחלק למקרים:

(1) כל הפעולות למעט דחיפות לא מרוות.
(כולל קידום מצביעים לקשת נוכחית, relabel, דחיפות מרוות) חסום ע"י למה 4.2 ע"י $O(nm)$, וזה חסום ע"י $O(n^3)$.

(2) דחיפות לא מרוות: אם מצומת יוצאת דחיפה לא מרווה, אז הוא מפסיק להיות פעיל, ולכן בכל מעבר יש לכל היותר דחיפה לא מרווה אחת מכל צומת.
מכאן שמס' הדחיפות הלא מרוות הוא $O(n^3)$.

מסקנה: זמן הריצה של האלג' כאשר Q מנוהל בשיטת FIFO הוא $O(n^3)$.

שיטות נוספות:

- וריאציות שבהן חוזרים על push/relabel(v) מס' כלשהו של relabels או עד ש- $e(v)=0$.
- בחר צומת ב-Q בעל תווית גבוהה ביותר.
- wave method.

ניתן להפחית את זמן הריצה ל $O\left(n \cdot m \lg \frac{n^2}{m}\right)$ ע"י שימוש במבני נתונים מתוחכמים.