אלגוריתמים 1

תוכן העניינים

פרק 1

אלגוריתמי BFS ו-DFS

1. BFS - Breadth First Search

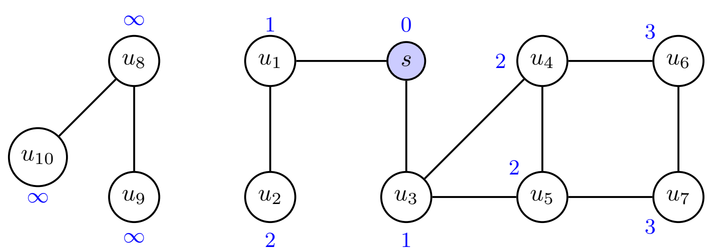
שאלה 1.1 כיצד לחשב מסלול קצר ביותר בין שני צמתים בגרף לא מכוון G?

1.1. הגדרת המרחק בגרף לא מכוון.

הגדרה 1.1 (המרחק בין צמתים u,v בגרף G)

בהינתן גרף לא מכוון $G=(V,E)$ ושתי צמתים $u,v\in V$,

המרחק בין u ו-v ב-G הוא האורך (מספר קשתות) של המסלול הקצר ביותר בין u ו-v ב-G. נסמן מרחק זה ב-$\delta_G(u,v)$ או ב-$\delta(u,v)$.



איור 1: המרחקים $\delta(s,u)$ מצומת s בגרף לא מכוון זה מסומנים בכחול ליד כל צומת $u\in V$.

טענה 1.1 (המקבילה לאי-שוויון המשולש)

יהי $G=(V,E)$ גרף לא מכוון, ויהי $s\in V$. לכל קשת $e=(u,v)\in E$ מתקיים:

$$\underbrace{\delta(s,v)}_{\text{המרחק בין } s \text{ ו-} v} \leq \underbrace{\delta(s,u)}_{\text{המרחק בין } s \text{ ו-} u} + \underbrace{1}_{\text{אורך הקשת } e}$$

הוכחת הטענה. אם אין מסלול בין u ו- s ב-G אז $\delta(s,u)=\infty$ והטענה מתקיימת.

אחרת, יהי P מסלול קצר ביותר בין s ו-u ב- G, כאשר אורכו שווה ל-$\delta(s,u)$.
נשרשר ל-P את הקשת e, וקיבלנו מסלול ב-G בין s ו-v, שאורכו שווה ל-$\delta(s,)+1$.

$\delta(s,v)$ שווה לאורך המסלול הקצר ביותר בין s ו-v ב-G, ולכן $\delta(s,v)\leq\delta(s,u)+1$. ■

1.2. מוטיבציה לאלגוריתם BFS. נרצה לחשב את המרחק בין צומת s לכל צומת בגרף:

- קלט: גרף לא מכוון $G=(V,E)$ וצומת $s \in V$.
- מטרה: לחשב לכל $v \in V$ את $\delta_G(s,v)$.

האינטואיציה: להתחיל מהצומת היחיד שעבורו יודעים את $\delta(s,?)$, וזה s עצמו.

1.3. אלגוריתם ה-BFS.

- אתחול: $\lambda(v) \leftarrow \begin{cases} 0 & v=s \\ \infty & v\neq s \end{cases}$, $T \leftarrow \{s\}$, $Q \leftarrow \{s\}$ תור.
- כל עוד $Q \neq \emptyset$:
  (1) יהי u הצומת בראש התור Q.
  (2) לכל קשת $e=(u,v)\in E$ כך ש-$v \notin T$:
    (א) $T \leftarrow T \cup \{v\}$.
    (ב) $\lambda(v) \leftarrow \lambda(u)+1$.
    (ג) הכנס את v לסוף התור Q.
  (3) הוצא את u מהתור Q.

1.4. נכונות האלגוריתם.

הערה 1.1 (סימון מקובל בקורס) עבור גרף $G=(V,E)$, נסמן $|V|=n,\ |E|=m$.

שאלה 1.2

(1) מדוע האלגוריתם מחזיר תשובה נכונה?
(2) עד כמה האלגוריתם יעיל? (בד"כ יעילות תתייחס לזמן)

נתחיל מ-(2).

- האתחול: $O(n)$.
- האיטרציה בה u יוצא מ-Q: $O(\deg(u))$.

כמו כן,

- כל צומת נכנס ל-Q לכל היותר פעם אחת.
- כל צומת שנכנס ל-Q גם יוצא מ-Q.

סך הכל זמן ריצה:

$$\underbrace{O(n)}_{\text{אתחול}}+\underbrace{O\left(\sum_{u\in V}\deg(u)\right)}_{\text{חסם עליון על זמן הריצה של כל האיטרציות}}=O(n+m)$$

■

נתמקד בטענה (1), ונוכיח אותה תוך שימוש בטענות העזר הבאות:

למה 1.1 ("λ לא מפספס למטה") יהי $G=(V,E)$ גרף לא מכוון, ותהא צומת $s\in V$. יהיו $\forall v\in V,\ \lambda(v)$ הסימונים שהתקבלו מריצת BFS על G החל מ-s. אזי:

$$\lambda(v)\geq\delta(s,v)\,,\ \forall v\in V$$

הוכחה. יהי $v\in V$.

אם v לא נכנס ל-Q, יתקיים $\lambda(v)=\infty$ והטענה נכונה.

אם v נכנס ל-Q (וזה קורה בדיוק פעם אחת), נוכיח את הטענה באינדוקציה על סדר כניסת הצמתים ל-Q:

- בסיס: s נכנס ראשון לתור (המקרה ש-$v=s$), ואז:

$$\lambda(s)=\underbrace{0}_{\text{הגדרת האלגוריתם}}=\delta(s,s)$$

- צעד: נניח נכונות עבור k הצמתים הראשונים שהוכנסו לתור, נניח כי v היא הצומת ה-$k+1$ שהוכנסה לתור.

ברגע ההכנסה של v ל-Q, נסמן ב-u את הצומת שבראש Q, ונקבל:

$$\lambda(v)\underbrace{=}_{\text{הגדרת האלגוריתם}}\lambda(u)+1\underbrace{\geq}_{\substack{\text{הנחת אינדוקציה}\\ u \text{ עבור}}}\delta(s,u)+1\underbrace{\geq}_{\substack{\text{אי שוויון המשולש}\\ (u,v)\in E\text{-ו } s \text{ עבור}}}\delta(s,v)$$

■

למה 1.2 יהי $(v_1,v_2,\ldots,v_k)$ תוכן Q בשלב כלשהו של ריצת BFS על G החל מ-s. אזי:

(1) $\lambda(v_1)\leq\lambda(v_2)\leq\ldots\leq\lambda(v_k)$.

(2) $\lambda(v_k)\leq\lambda(v_1)+1$.

הוכחה. נוכיח באינדוקציה על סדר הפעולות של הכנסה/הוצאה מ-Q:

- בסיס: האתחול הוא כש-Q מכיל רק את s. לכן (1) ו-(2) מתקיימים באופן ריק.
- צעד: נניח נכונות עבור r הפעולות הראשונות, ונוכיח עבור הפעולה ה-$r+1$.

אם הפעולה ה-$r+1$ הייתה הכנסה, נניח שהכנסנו את v ו-u בראש התור, אז:

$$\lambda(v)=\lambda(u)+1$$

לפי הגדרת האלגוריתם.

בגלל שלפני הוספת v ל-Q (1) ו-(2) התקיימו, זה יתקיים גם לאחר הוספת v.

אם ההפעלה ה-$r+1$ הייתה הוצאה, אז ברור שמהנחת האינדוקציה (1) ו-(2) יתקיימו גם לאחריה.

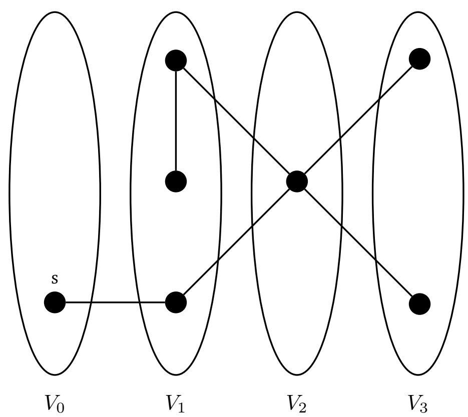
■

משפט 1.1 (הוכחת נכונות אלגוריתם BFS)
יהי $G=(V,E)$ גרף לא מכוון ו-$s\in V$.
אז בסיום ריצת BFS על G החל מ-s מתקיים:

$$\forall v\in V,\ \ \lambda\left(v\right)=\delta\left(s,v\right)$$

הוכחת המשפט משתמשת בטענות 1 ו-2.
רעיון ההוכחה: נסתכל על שכבות הגרף לפי מרחק מ-s:

$$V_k\triangleq\left\{u\in V:\delta\left(s,u\right)=k\right\}$$



איור 2: שכבות של גרף לא מכוון לדוגמה עבור צומת s כלשהי

הוכחת המשפט. נניח שב-G אין מסלול בין s ו- v $\Longleftarrow$ $\delta\left(s,v\right)=\infty$.
לפי טענה 1 נקבל ש-$\lambda\left(v\right)\geq\infty$, כלומר $\lambda\left(v\right)=\infty$, והמשפט נכון.

נניח שב-G יש מסלול בין s ו-v ונסמן $\delta\left(s,v\right)=k$.
נוכיח את המשפט באינדוקציה על k:

- בסיס: $k=0$, אז $v=s$, והמשפט מתקיים מפני שבאתחול מוגדר $\lambda\left(s\right)=0$.
- צעד: נניח כי $v\in V_k$, ונסמן:

$$A\triangleq\left\{u\in V_{k-1}|(u,v)\in E\right\}$$

כאשר הגדרת A אינה תלויה באלגוריתם.
נסמן ב-u^* את הצומת ב-A שהיא הראשונה לצאת מהתור Q.

נשים לב ש-A אינה יכולה להיות ריקה, ולפי הנחת האינדוקציה,
בסיום ריצת האלגוריתם לכל הצמתים ב-A ישנו ערך λ השווה ל-$k-1$,
ולכן בהכרח כל אחד מהם הוכנס לתור Q.

נראה שבאיטרציה שבה u^* נמצא בראש התור Q, לצומת v מתקיים ש-$\lambda\left(v\right)=\infty$
(כלומר, v עדיין "לא התגלה").

נניח בשלילה שזה לא המצב, ולכן יש איטרציה קודמת לזו ש-u^* נמצא בה בראש Q, שבה v מוכנס לתור Q (ונניח ש-w נמצא בראש התור Q באיטרציה זו).

בגלל בחירת u^*, מתקיים ש-w הוא שכן של v בשכבה j, כך ש-$0\leq j\leq k-1$ (נובע מלמה 1.2).

לפי הנחת האינדוקציה $\lambda\left(w\right)<\lambda\left(u^*\right)$, וכעת:

$$\lambda\left(v\right)\underbrace{=}_{\text{הגדרת האלגוריתם}}\lambda\left(w\right)+1<\lambda\left(u^*\right)+1\underbrace{=}_{\substack{\text{הנחת האינדוקציה}\\ \text{עבור } u^*}}\left(k-1\right)+1=k=\delta\left(s,v\right)$$

סה"כ קיבלנו $\lambda\left(v\right)<\delta\left(s,v\right)$, וזו סתירה מלמה 1.1.

באיטרציה שבה u^* בראש התור Q, הצומת v מקיימת $\lambda\left(v\right)=\infty$, ולכן באיטרציה זו v יקבל סימון $\lambda\left(v\right)=k$ ויוכנס ל-Q.

■

2. DFS - Depth First Search

<u>משימה</u>: למצוא רכיבים קשירים היטב של גרף מכוון בזמן לינארי.

2.1. חותמות זמן: זמני גילוי וסיום של צומת (במהלך אלגוריתם סריקה).

הגדרה 1.2 $s\left(u\right)$ - זמן הגילוי של צומת u

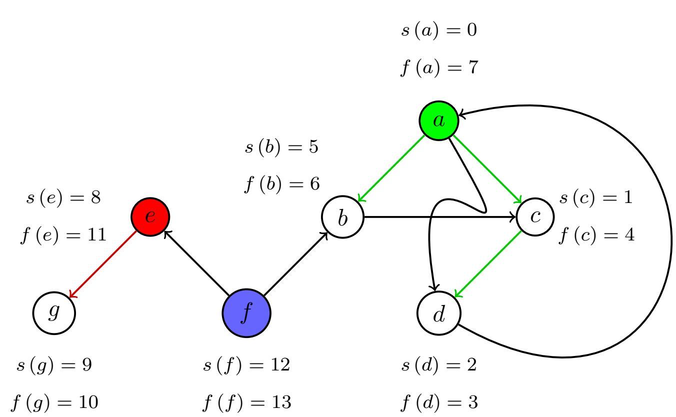
הגדרה 1.3 $f\left(u\right)$ - זמן סיום של צומת u.

2.2. האלגוריתם.

- <u>אתחול</u>:
  (1) $\forall u\in V,\ \text{status}\left(u\right)\leftarrow\text{unvisited}$
  (2) $\forall u\in V,\ \begin{matrix}p\left(u\right)\leftarrow\text{NULL}\\ t\leftarrow0\end{matrix}$
- כל עוד יש צומת u כך ש-$\text{status}\left(u\right)=\text{unvisited}$: בצע $\text{visit}\left(u\right)$.
- <u>$\text{visit}\left(u\right)$</u>:
  (1)
  - $s\left(u\right)\leftarrow t$
  - $t\leftarrow t+1$
  - $\text{status}\left(u\right)\leftarrow\text{visited}$

  (2) לכל קשת $\left(u\rightarrow v\right)\in E$, אם $\text{status}\left(v\right)=\text{unvisited}$, אז $p\left(v\right)\leftarrow u$, וגם $\text{visit}\left(v\right)$.

  (3) $\begin{cases}f\left(u\right)\leftarrow t\\ t\leftarrow t+1\end{cases}$



איור 3: דוגמת הרצה של אלגוריתם DFS.

מסקנה 1.1 בריצת DFS על גרף מכוון G, לככל צומת $u \in V$,
$\text{visit}\left(u\right)$ יקרא בדיוק פעם אחת.

2.3. זמן ריצה.

- מה זמן הריצה של אלגוריתם ה-DFS?
- עבור צומת $u \in V$, כמה זמן לוקח לבצע $\text{visit}\left(u\right)$ ללא הקריאות הרקורסיביות (אם יש) שנובעות ממנו? - $O\left(1\right)+O\left(\deg_{\text{out}}\left(u\right)\right)$.

$\Longleftarrow$ סה"כ $O\left(n+m\right)$ (ובפרט האלגוריתם עוצר).

הערה 1.2 לאלגוריתם ה-DFS דרגות חופש רבות.
חותמות הזמן s,f מהוות תיעוד של היסטוריית ריצת האלגוריתם.

הגדרה 1.4 (יער ה-DFS) נסתכל על הגרף $G_p=\left(V,E_p\right)$, כאשר:

$$E_p=\left\{\left(p\left(v\right)\rightarrow v\right)\in E:p\left(v\right)\neq\text{NULL}\right\}$$

נשים לב ש-G_p הוא תת-גרף של G.

משפט 1.2 (תרגיל) G_p הוא יער מכוון אשר פורש את כל צמתי V.

2.4. סוגי קשתות ביער ה-DFS.

שאלה 1.3 כיצד ניתן לסווג את קשתות G
בהינתן ריצה מסוימת של DFS?

הגדרה 1.5 (קשת עץ) $\left(u\rightarrow v\right)\in E$ היא קשת עץ, אם $p\left(v\right)=u$.

הגדרה 1.6 (קשת קדמית) $\left(u\rightarrow v\right)\in E$ היא קשת קדמית, אם אינה קשת עץ,
ובנוסף u אב קדמון של v ביער ה-DFS.

הגדרה 1.7 (קשת אחורית) $\left(u\rightarrow v\right)\in E$ היא קשת אחורית, אם u צאצא של v ביער ה-DFS.

הגדרה 1.8 (קשת חוצה) כל שאר הקשתות מכונות קשתות חוצות.

הערה 1.3 כאשר מבצעים DFS על גרף לא מכוון, יווצרו רק קשתות עץ וקשתות אחוריות (ללא הוכחה).

2.5. אפיון יחסי אב-צאצא ביער ה-DFS.

משפט 1.3 לכל גרף מכוון G ולכל ריצת DFS,
v צאצא של u ביער ה-DFS, **אם ורק אם** בזמן גילוי u, יש ב-G מסלול מ-u ל-v שכל הצמתים בו הן unvisited (פרט ל-u עצמו).

למה 1.3 לכל גרף מכוון G, לכל ריצת DFS ולכל $u, v \in V$,
בדיוק אחד משלושת הבאים מתקיים:

(1) $[s(u), f(u)]$ ו-$[s(v), f(v)]$ זרים, ו-u אינו צאצא של v ו-v אינו צאצא של u.
(2) $s(v) < s(u) < f(u) < f(v)$, ו-$u$ צאצא של v.
(3) $s(u) < s(v) < f(v) < f(u)$, ו-$v$ צאצא של u.

הוכחה. נניח $s(u) < s(v)$ (המקרה ההפוך - סימטרי).

- מקרה ראשון: $\boxed{s(v) < f(u)}$

נרצה להראות שאנחנו במקרה ג'.
ברגע גילוי v, עדיין לא סיימנו את $\text{visit}(u)$ (בגלל ש-$s(v) < f(u)$).

$\Leftarrow$ $\text{visit}(v)$ נקרא מתוך שרשרת קריאות רקורסיביות מתוך $\text{visit}(u)$.
$\Leftarrow$ $\text{visit}(v)$ מסתיים לפני $\text{visit}(u)$.
$\Leftarrow$ $f(v) < f(u)$
$\Leftarrow$ $\boxed{s(u) < s(v) < f(v) < f(u)}$.

מדוע v הוא צאצא של u?
נוכיח באינדוקציה לפי מספר הקריאות של visit שבוצעו בין $\text{visit}(u)$ ל-$\text{visit}(v)$.
בסיס: $\text{visit}(v)$ בוצע ישירות מתוך $\text{visit}(u)$.
לפי הגדרת האלגוריתם, $p(v) = u$, ולכן v צאצא של u.

צעד: נניח כי $\text{visit}(v)$ נקרא מתוך $\text{visit}(w)$.
$\Leftarrow$ $p(v) = w$, כלומר v הוא ילד ישיר (צאצא) של w.
לפי הנחת האינדוקציה, w הוא צאצא של u, ולכן v צאצא של u.

- מקרה שני: $\boxed{f(u) < s(v)}$

נרצה להראות שאנחנו במקרה א'.
חייב להתקיים:

$$s(u) < f(u) < s(v) < f(v)$$

מכיוון שלא ניתן לסיים צומת לפני שמגלים אותו.

נראה ש-v אינו צאצא של u (המקרה ההפוך - סימטרי):

אם נניח בשלילה ש-v הוא כן צאצא של u, אז צריך להתקיים ש-$\text{visit}\left(v\right)$ מתרחש בשרשרת קריאות רקורסיביות שמקורן ב-$\text{visit}\left(u\right)$, ובפרט $\text{visit}\left(v\right)$ מסתיים לפני סיום $\text{visit}\left(u\right)$, ז"א $f\left(u\right) > f\left(v\right)$ - בסתירה!

■

מסקנה 1.2 (מטענת העזר)

v צאצא של u $\iff$ $s\left(u\right) < s\left(v\right) < f\left(v\right) < f\left(u\right)$.