11/11/08

1

אלגוריתמים - שיעור 2

המשך ניתוח האלגוריתם BFS

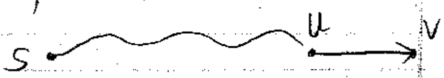
$\delta(s,u)$ = אורך (בקשתות) של המק"ב מ-s ל-u.

$d[u]$. ראינו את התכונות הבאות:

1) אם (u,v) קשת אז $\delta(s,v) \le \delta(s,u)+1$

2) אם (u,v) הקשת האחרונה במק"ב מ-s ל-u אז

$\delta(s,v) = \delta(s,u)+1$



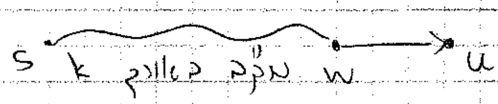
<u>למה</u>: אם u נגיש מ-s, האלג' מבקר ב-u (וגם הכיוון השני מתקיים, טריו')

<u>הוכחה</u>: באינדוקציה על $k=\delta(s,u)<\infty$

<u>k=0</u>: $u=s$ ובוודאי מבקרים בו

אז נתון ל k-1 ו-u מקיים $\delta(s,u)=k$, w הקודקוד הקודם ל-u

האלגוריתם ביקר ב-u זה מכיוון ש-u שכן לצומת [illegible] וקודקוד של s



לפי תכונה 2) $\delta(s,w)=k-1$ ולפי הנחת האינדוקציה האלג' מבקר ב-w ואז עובר על כל שכניו, כולל u

<u>טענה</u>: בכל רגע, בתור Q נמצאים צמתים עם לכל היותר 2 ערכי d עוקבים בסדר d עולה.

<u>הוכחה</u>: באינדוקציה על צעדי האלגוריתם.

בהתחלה Q מכיל רק את s עם ערך $d=0$.

נניח שהטענה נכונה עד רגע מסוים

$Q = V_1 \; V_2 \ldots \ldots V_m$

$d = i \; i \ldots i \; i+1 \ldots i+1$

האלגוריתם מוציא את V_1 מהתור, עובר על כל שכניו ונותן לחדשים מביניהם ערך $d=i+1$ ושם אותם בסוף התור,

ולכן התכונה נשמרת.

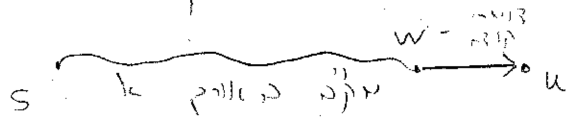
<u>משפט</u>: לכל צומת u $d[u]=\delta(s,u)$ (כולל האפשרות ששניהם ∞)

<u>הוכחה</u>: אם $d[u]=\infty$ בסיום $\Longleftrightarrow$ האלג' לא ביקר ב-u $\Longleftrightarrow$ u לא נגיש מ-s $\overset{\text{לפי הגדרה}}{\Longleftrightarrow}$ $\delta(s,u)=\infty$.

לשאר הצמתים נניח באינדוקציה על $k=\delta(s,u)$.

<u>$k=0$</u> $u=s$, $d[s]=0$ באתחול , $\delta(s,s)=0$

נניח נכונות ל $k-1$ ויהא u צומת במרחק k מ-s



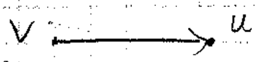
כמו קודם $\delta(s,w)=k-1$ ולכן לפי הנחת האינדוקציה האלג' ביקר ב-w ושם $d[w]\leftarrow k-1$

מהטענה על Q נובע שהאלג' מבקר בצמתים לפי סדר d עולה. ולכן מספיק שנראה ש $d[u]$ גדול מ-$d[w]$. מזה נובע ש-u עדיין לא בתור כש-w מוצא מהתור.[* ולכן u יהיה שכן חדש של w (כש-w יגיע לראש התור) (שהאלגוריתם עוד לא נתקל בו) ואז האלג' ישים ב $d[u]\leftarrow d[w]+1=k$]

* המרצה מחק

כל עוד w בתור יש בתור (אחרי w) רק צמתים עם ערכי $d=k, k-1$ (ואולי $k-2$ אבל כאלה רק לפני w). אם u טופל מקודם אז $d[u]=k$ והוכחנו. אחרת אז $\rightarrow$ כש-w מגיע לראש התור u לא בתור $\rightarrow$ u שכן חדש של w והאלג' ישים בו $d[u]\leftarrow d[w]+1=k$.

כמו כן נוכיח ש $d[u]$ גדול מ-$d[w]$: לכל u $d[u]\geq\delta(s,u)$ וגם את זה ניתן להוכיח באינדוקציה.



$$d[u]\leftarrow d[v]+1 \geq \delta(s,v)+1 \geq \delta(s,u)$$

לפי טענה 1) לפי אינדוקציה

ולכן $d[w]=\delta(s,w)=k-1<k=\delta(s,u)\leq d[u]$.

<u>שימושים</u>: מנה את שאלת של *נגישות מ-s, למשל האם צומת u נגיש מצומת s? הרצת BFS מ-s או מציאת כל הצמתים הנגישים מ-s וגם *מציאת מרחק כל צומת מהר מ-s (מערכת שדה ה-d). *מציאת "המסלולים" מ-s לכל צומת נגיש (מערך שדה ה-π).

שימו לב המסלולים באופן מפורש יכולה לקחת $\Theta(|V|^2)$ זמן.

11.11.08 (3)

(למשל את הגרף נראה כך

$s \to v_1 \to v_2 \to v_3 \to v_4 \cdots \to v_n$

לכן נשתמש בייצוג מערך ע"י π:

כדי למצוא מסלול מ-s ל-u נלך בעקבות מצביעי π

$v_1 \leftarrow u$

$v_2 \leftarrow \pi[v_1]$

$v_3 \leftarrow \pi[v_2]$

$\vdots$

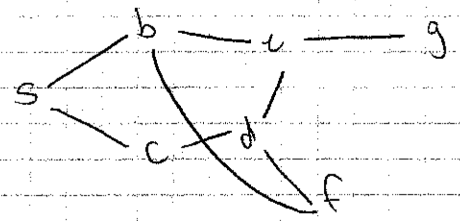
עד שנגיע ל-s

והמסלול יהיה רשימת הצמתים ההפוכה.

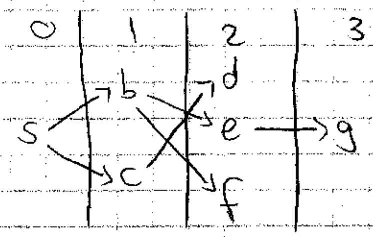
ה-BFS מסדר את הצמתים הנגישים מ-s בשכבות layers.

בשכבה ה-i נמצאים הצמתים עם $d[v]=i$, $i=0,1,\ldots,n$

למשל:



השכבות נראות כך:

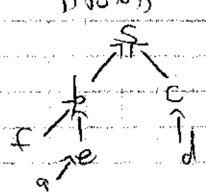


(הקשתות האחרונות: של-d ו-ל-f הן קשתות שבהן האלגוריתם לא עשה כלום שכן הגיע כבר לצמתים e ו-f)

אם הגרף מכוון, כל קשת שאינה "בסיסית" —> אלה ש-BFS השתמש בהן לבניית d מחברת צומת בשכבה i לצומת כלשהי בשכבה $\leq i+1$ (בדוגמא שלנו יש דוגמאות לשתי הקשתות מאותה שכבה, אך הקשת יכולה גם ללכת אחורה)

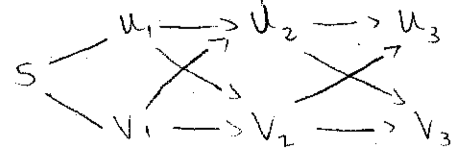
אם הגרף לא מכוון, כל קשת מחברת צמתים באותה שכבה או בשכבות סמוכות

הערה: מצביעי π מגדירים למעשה עץ מכוון ששורשו s שנקרא לו עץ "המסלולים". מסלול לצומת u בעץ = המסלול מהשורש ל-u



אלגוריתמים
11.11.98 שיעור 2 (4)

הערה: לא כל המק"בים מתקבלים בדרך זו



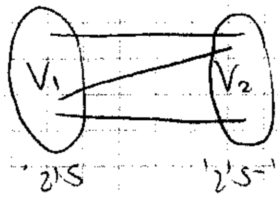
אפשר להגיע אל כל המק"בים ע"י רק שכל צומת בשכבה i לשני יצביע על כל הצמתים משכבה $i-1$ שיש קשת מהם אליו.

שימוש נוסף: נתון גרף לא מכוון G. לקבוע אם G הוא דו-צדדי ואם כן למצוא פרוק מתאים של V.

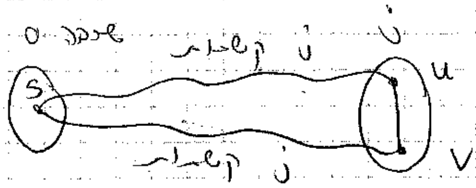
גרף לא מכוון הוא דו צדדי $\iff$ אין בו מעגל באורך אי-זוגי.

הוכחה: $\Rightarrow$: אם G דו-צדדי אז כל קשת עוברת אלטרנטיבית מ-V_1 ל-V_2 או להפך ולכן בכדי לחזור לאותו צומת צריך מס' זוגי של קשתות.

(כל קשת עוברת מ-V_1 ל-V_2 או מ-V_2 ל-V_1)



$\Leftarrow$: נוכל להניח ש-G קשיר. אחרת, נפעיל את האלגוריתם לכל רכיב קשירות בנפרד. ניקח צומת s כלשהו ונריץ BFS ממנו. נקבל את G הצמתים מסודרים בשכבות ונראה שאין קשתות שמחברות 2 צמתים באותה שכבה, כי אחרת היינו נקבל מעגל אי-זוגי.



סך הכל $2i+1$ קשתות

ולכן כל הקשתות מחברות שכבות סמוכות. ולכן נגדיר

V_1 = כל הצמתים בשכבות הזוגיות (כולל s)

V_2 = כל הצמתים בשכבות אי-זוגיות

(כלומר בזמן לינארי נגלה האם הגרף דו-צדדי)

יומם
11.11.08 שיעור 2
5

חיפוש לעומק - Depth-first Search DFS

(צומת התחלה s)

מבקר בכל צמתי הגרף

(ב- BFS כשהאלגוריתם מגיע לצומת חדש הוא מבקר בכל שכניו ולא מתקדם קדימה כמו ב- DFS כלומר החיפוש לרוחב, ואילו ב- DFS רק כשהאלג' יתקע יחזור אחורה)

DFS מנהל את החיפוש ע"י כך שהוא בוחר כל פעם צומת התחלה s חדש שעדיין לא ביקרנו בו ומתחיל ממנו חיפוש חדש.

DFS-VISIT הינו רקורסיבי ומבצע חיפוש בניסיון להגיע לכל מה שניתן מצומת ההתחלה הנוכחי s שעדיין לא ביקרנו בו.

באלגוריתם יש צבעים לכל צומת u:

לבן -> עוד לא נתקלנו ב- u

אפור -> נתקלנו אך טרם סיימנו

שחור -> סיימנו את הביקור ב- u

שימור - מונה (שמתאר את סדר האירועים שהאלגוריתם נתקל בו)

בכל פעם שאנו או נתקלים בצומת חדש או מסיימים טיפול בצומת.

שדות: $d[u]$ (discovery) = הזמן בו נתקלנו ב- u לראשונה

$f[u]$ (finish) = הזמן בו סיימנו את הטיפול ב- u.

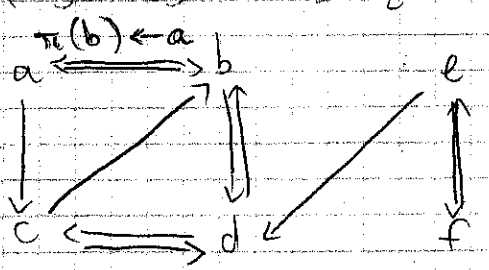
$f[u] > d[u]$

$\pi[u]$ = הצומת הקודם שגילה את u

DFS משתמש ברשימה של הצמתים (בסדר כלשהו)

הקלט G גרף מכוון או לא נתון ע"י רשימת שכנות (כך הן בסדר כלשהו, סדר אחר יגרום לאלגוריתם להתנהג שונה)

דוגמאת הרצה:



נניח ש- a הצומת הראשון ש- DFS בוחר. ל- a שני שכנים, נניח שמסתכלים על b. b כרגע לבן, יש לו שכן אחד והוא חדש, נעבור אליו. מ- b נעבור ל- c ומ- c ל- d. d כרגע

11.11.08 שיעור 2 (6)

אפור, לכן לא משנה דבר. כעת נצבע את c בשחור, שכן אין לו שכנים לבנים ונחזור בקריאה הרקורסיבית.
הבנים של d מגיעים רק ל- c ונתקע לכן נבחר צומת חדש.
הרצה מהפרוצדורה הראשית יצאה כל הצמתים כבר עוברו לצבע
האלגוריתם יסתיים.

	d[ ]	f[ ]
a	1	8
b	2	7
c	4	5
d	3	6
e	9	12
f	10	11

מצביעי π יוצרים אוסף של עצים מכוונים לכיוון השורש (יער)

$$c \to d \to b \to a \qquad f \to e$$

<u>עקרון הקינון</u> (nesting)

לכל צומת u מתאים אינטרוול זמן בו u חי/פעיל [d[u]...f[u]]
לכל שני צמתים u,v או שיש להם אינטרוולים זרים או שהאינטרוולים שלהם מקוננים זה בזה.

זרים: [d(u) ___ f(u)] [d(v) ... f(v)]

מקוננים: [d(u) _________ f(u)]
[d(v) ... f(v)]

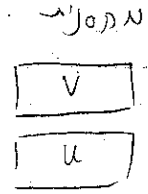
(לא יתכן מצב כזה)

ואם הם מקוננים אז הצומת v עם האינטרוול הקטן הוא צאצא של u בעץ ה- DFS.

<u>הוכחה / הסבר</u>: רקורסיה ⟵ מחסנית
בזמן d[u] מכניסים את u למחסנית
בזמן f[u] מוציאים את u מהמחסנית ויותר הוא לא חוזר לשם

11.11.09 שיעור 2 (7)

אם $u \leftarrow \pi[v]$ זה קורה בזמן קריאה רקורסיבית - u קורא ל-v



כלומר מצביע π הוא מצביע גבוה יותר במחסנית לצומת מתחתיו

ולכן לא יתכן מעגל המורכב ממצביעי π.

לכן π מגדיר גרף מכוון חסר מעגלים ולכל צומת דרגת יציאה 1

$\equiv$ יער מכוון לכוון השורשים.

<u>הוכחת הקינון</u>: נניח שהאינטרוולים אינם זרים. בה"כ $d(u)<d(v)<f(u)$

נראה שגם $f(v)<f(u)$

v מתגלה אחרי ש-u התגלה אבל לפני ש-u מסיים, ולכן כש-v התגלה u במחסנית ו-v נכנס לראש המחסנית מעל u,

ולכן v יוצא מהמחסנית לפני u כלומר $f(v)<f(u)$ כנדרש.

במקרה המקונן v נמצא במחסנית מעל u ומבנה המחסנית מקביל לזה בעץ π,

v צאצא של u $\longleftrightarrow$ (שקול) u אב קדמון של v.

$u \leftarrow w_1 \leftarrow w_2 \leftarrow w_3 \leftarrow \dots \leftarrow w_{k-1} \leftarrow v$

v
w_{k-1}
w_3
w_2
w_1
u

<u>סיווג קשתות הגרף ע"י DFS</u>

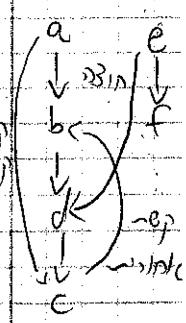
ה-DFS מבקר את קשתות הגרף. בחלק מהן הוא משתמש להגדרת π לבניית עץ ה-DFS (קשתות עץ) ובשאר לא.

קשת קדמית = מחברת אב קדמון לצאצאו (forward)

למשל הקשת מ-a ל-c

קשת אחורית = מצאצא לאב קדמון (או לאב עצמו) back edge

קשת חוצה = מחברת שני צמתים שאינם צאצא/אב קדמון cross



<u>G מכוון</u>

מסתכלים על קשת (u,v) ברגע שבו u בודק אותה.

1) אם v לבן זו קשת עץ,

2) אם v אפור זו קשת אחורית

3) אם v שחור זו קשת קדמית/חוצה

11.11.08 שיעור 2 8

2) ! ברגע זה v כבר התגלה אך לא הסתיים. ברגע זה u

בראש המחסנית. u במחסנית מתחת ל-u

<-- u צאצא של v. לכן זו קשת אחורה.

3) v שחור זמן נוכחי < $f(v)$

$d(u) <$ זמן נוכחי $< f(u)$

לפי עקרון הקינון יש 2 אפשרויות:

1) $f(v) < d(u)$ <-- האינטרוולים זרים כלומר u,v אינם צאצאים / $d(v)<d(u)$

אב קדמון והקשת חוצה.

כיון הקשת הולכת מצומת "מאוחר" יותר (u) לצומת "מוקדם" (v).

2) $d(u) < f(u)$ <-- הם מקוננים $d(u) \text{———} f(u)$ $d(u)<d(v)$

$d(v) \text{—} f(v)$

<-- v צאצא של u ולכן קשת קדמית.

* דרך אלגוריתמית היא לבדוק $d(u) : d(v)$