הרצאה 2

אלגוריתם חיפוש לעומק

דוגמה:

$\overbrace{\downarrow}^{\text{Start}}$	☻	$\rightarrow$	
$\downarrow\rightarrow$	$\rightarrow$	$\uparrow$ $\downarrow$	☻
	☻	$\rightarrow$	$\downarrow$
☻		☻	

רובוט סורק אזור לצורך גילוי מוקשים, ההתקדמות בכל צעד: ימינה שמאלה למטה, למעלה.

אלגוריתם DFS (Depth First Search)

מנסה להתקדם כמה שיותר לעומק הגרף.
כאשר נבקשר בצומת v, אם יש קשת (u,v) לצומת u שעוד לא "התגלה", נחצה את הקשת ונמשיך את החיפוש מהצומת u.
<u>המטרה:</u> יש לגלות את כל הצמתים בגרף.

DFS על גרף מכוון:

קלט: גרף מכוון $G=(V,E)$, צומת s.
פלט: לכל $v \in V$, $d[v]$, זמן הגילוי של v.
סימונים:

- $\star$ $d[v]$ זמן גילוי של v.
- $\star$ $\pi[v]$ הצומת שגרם ל-v להתגלות.

```
DFS:
1.For all v∈V  d[v] ← 0,π[v] ← null
     mark all edges "unused"
     i ← 0,  v ← s
.2i ← i+1,  d[v] ← i
3.While there are unused out-edges from v,
     choose unused edges (v,u), mark (v,u) as used
     if d[u]=0 :  {π[u] ← v, v ← u, i ← i+1, d[v] ← i}
4.If π[v] ≠ null then  v ← π[v] and go to (3)
     else if there is u∈V with d[u]=0
     then v∈u and go to (2).
5.stop
```

- ⋆ נשים לב כי בהרצות שונות של DFS נוכל לקבל פלטים שונים, אך הכולן נקבל "יער" שבו כל צומת מופיע מאיזשהו עץ מכוון.
- ⋆ בנוסף , DFS לא בהרכח מוצא מרחקים קצרים.
- ⋆ בסיום הרצת DFS נקבל predecessor subgraph. זהו תת-גרף שבו לכל צומת v מופיע קשת $(v,\pi(v))$ כפי שנמצא ע"י האלגוריתם, סימון: G_π.

For each $u \in V$ do:
  $\{$color$[u] \leftarrow$ white$, \pi[u] \leftarrow null\}$
For each $u \in V$ do :
  if color$[u]$ =white then DFS-VISIT(u(
DFS-VISIT(u):
  $color[u] \leftarrow gray$
  $i \leftarrow i+1$
  $d[u] \leftarrow i$
  For each $v \in Adj[i]$ do
    if color[v[=white then $\{\pi[v] \leftarrow u,$ DFS-VISIT$(v)\}$
  $i \leftarrow i+1$
  $f[u] \leftarrow i$

color$[u]$ - $\{white, gray\}$.
$f[u]$ - זמן היציאה האחרון מ-u.
$Adj[u]$ - אוסף השכנים של u.

זמן הריצה של DFS :

- ⋆ לולאת האתחול: $\theta(|V|)$.
- ⋆ נסמן ב-$T(\text{DFS-VISIT})$ את מספר הפעולות המבוצעות בקריאה ל-DFS-VISIT עבור הצומת .
  נשים לב כי DFS-VISIT נקראת בדיוק פעם אחת עבור v כאשר v "לבן", ומיד בכניסה לפרוצדורה v הופך ל-"אפור". בנוסף, מספר הפעולות בלולאת ה-"For" של DFS-VISIT הוא <u>לינארי</u> במספר השכנים של v. לכן סיבוכיות הקריאות ל-DFS-VISIT:

$$\sum_{v\in V} \text{T(DFS-VISIT)}= \sum_{v\in V}\theta(|Adj[v]|) = \theta(|E|)$$

$\Leftarrow$ סה"כ זמן הריצה של DFS: $\Theta(|E|+|V|)$.

תכונות של DFS:

1. <u>התכונה הבסיסית:</u> G_π הוא יער, שכן המבנה של עצי DFS מקשף את הקריאות הרקורסיביות ל-DFS-VISIT.
2. v הוא צאצא של u בעץ DFS אם v התגלה כאשר u היה אפור ולפני שנקבע ערך ל-$f[u]$.
3. <u>תכונת הסוגריים:</u> נייצג את הגילוי של צומת u ע"י סוגר שמאלי $'(u'$ ואת סיום הטיפול בו ע"י $'u)'$.
   אזי, ההיסטוריה של "גילוי" ו"סיום הטיפול" מגדירה ביטוי שבו הסוגריים מקוננים היטב.

משפט 1 (הסוגריים):

בכל חיפוש לעומק של גרף מכוון/לא-מכוון G, לכל שני צמתים u ו-v מתקיים בדיוק אחד מהתנאים:

1. האנטרוולים $[d[u],f[u]]$ ו- $[d[v],f[v]]$ זרים לחלוטין ואין קשר של אב-קדמון/צאצא בין הצמתים.
2. האנטרוול $[d[u],f[u]]$ מוכל ממש בתוך $[d[v],f[v]]$ ו-u צאצא של v בעץ DFS.
3. $[d[v],f[v]]$ מוכל ממש ב-$[d[u],f[u]]$ ו-v צאצא של u בעץ DFS.
4. תכונה נוספת של צאצאים ביער במשפט הבא.

משפט 2 (<u>המסלול הלבן</u>):

ביער DFS של גרף (מכוון/לא-מכוון) צומת v הוא צאצא של צומת u אם"ם בזמן $d[u]$, הזמן בו u התגלה,ניתן להגיע ממנו ל-v ע"י מסלול המורכב כולו מצמתים לבנים.

הוכחה:

$\Leftarrow$

נניח ש-v צאצא של u. יהיה w צומת על המסלול בין u ו-v בעץ DFS כך ש-w צאצא של u. ממשפט 1, $d[u]<d[w]$, לכן w היה לבן בזמן $d[u]$.

$\Rightarrow$

נניח בשלילה שיש מסלול לבן מ-u ל-v בזמן $d[u]$, אבל v לא נהיה צאצא של u בעץ DFS.
נניח ש-v הוא הצומת הראשון על המסלול הלבן שאנינו צאצא של u.
יהיה w הצומת לפני v על המסלול הלבן כך ש-w צאצא של u (או $w = u$).
אזי ממשפט 1 $f[w] \leq f[u]$.
נשים לב כי v מוכרח להתגלות אחרי u, אבל לפני שנצא בפעם האחרונה מ-w דהיינו:

$$d[u] \underbrace{<}_{\text{at d[u] there is a white path to v}} d[v] \underbrace{\leq}_{\text{we won't finish with w before we get to v}} f[u]$$

ממשפט 1 נקבל כי $[d[v], f[v]]$ חייב להיות מוכל ממש ב- $[d[u], f[u]]$
ולכן v יהיה צאצא של u בעץ DFS- סתירה ■.