

به نام یگانه معبود بخشنده مهربان

طراحی الگوریتم ها

Design and Analysis of Algorithms

گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه اصفهان

ترم اول سال تحصیلی ۹۰-۹۱

ارائه دهنده: پیمان ادیبی

روش های جستجو در گراف

Graph Search Methods

مفاهیم

■ گرافها در مسایل:

□ صریح

□ غیر صریح

■ اغلب نیاز داریم که درون گراف جستجو انجام دهیم.
■ برای جستجوی کارآمد در گرافها نیاز به روشهایی برای انجام کارهای زیر داریم:

□ علامت گذاری گره هایی که قبلا بررسی شده اند

□ تعیین اینکه کدام گره باید در قدم بعد بررسی شود

□ اطمینان یافتن از اینکه هر گره در گراف میتواند ملاقات شود

■ الگوریتم جستجو در گراف باید ساختار گراف را رعایت کند. یعنی هر گره جدید که بررسی میشود باید حداقل مجاور با یکی از گره هایی که قبلا بررسی شده باشد.

جستجوی عمق اول (Depth First Search)

- با داشتن گراف همبند $G(V,E)$ ، یک درخت جستجوی $T(V,F)$ (که درختی ریشه دار، جهت دار و پوشاست) و برجسبهای یکه k ($k=1,\dots,|V|$) که ترتیب جستجو را بیان میکنند، بدست می آورد (برجسب ریشه همیشه 1 است).
- الگوریتم بازگشتی (با فرض بدون جهت بودن G):

```
DFS(node v, int l, SearchTree T) {  
    label[v]=l;    l++;  
    for (each (v,u) in E)  
        if (label[u]==0) {  
            add edge (v,u) to T;  
            DFS(u, l, T);  
        }  
}
```

متغیرهای سراسری: $label, V, E$
فراخوانی سطح بالا در تابع اصلی:
graph $G(V,E)$;
set $F=\emptyset$;
SearchTree $T(V,F)$;
 $label[1..|V|]=zeros[1..|V|]$;
node $v = root(V)$;
DFS($v,1,T$);

جستجوی عمق اول (Depth First Search)

■ مثال: ...

■ زمان اجرای DFS در هر حالت $O(|E|)$ است، چون هر یال از گراف تنها یک بار بررسی میشود.

■ اگر G یک گراف جهتدار باشد، ممکن است بعضی از گره ها از یک گره بعنوان ریشه قابل دستیابی نباشند. در این حالت فراخوانی سطح بالا را بشکل زیر تغییر میدهیم:

```
int label[1..|V|]=zeros[1..|V|];
```

```
for (each v in V)
```

```
    if (label[v]==0)
```

```
        DFS(v, l, T);
```

■ مشکل DFS: ممکن است به جستجوی بی پایان در گرافهای
ضمنی منجر شود ...

جستجوی پهنا اول (Breath First Search)

■ الگوریتم: **BFS(node v, int l, SearchTree T){**

set M=∅, CL=∅, NL=∅; //The sets of expanded, current level, and next level nodes

int label[1..|V|] = zeros[1..|V|]; l=1; CL={v};

unmark all nodes;

■ مثال: ...

mark v;

while (CL != ∅){

for (each v in CL){

M=the set of unmarked neighbors of v;

mark all nodes in M;

NL = NL ∪ M;

for (each u in M)

add edge (u,v) to T;

if (label(v)==0){

label(v)=l; l++; }

}

CL = NL; NL = ∅;

}

}

■ **BFS** مشکل مذکور
برای **DFS** (افتادن در
حلقه بی پایان) را ندارد.