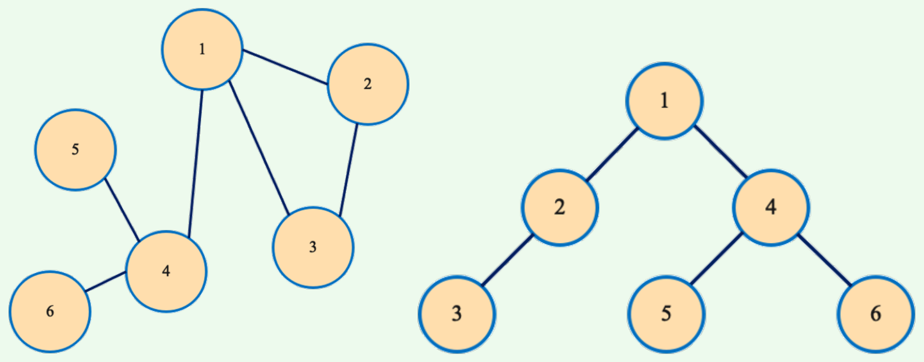
در جلسه‌ی قبل در مورد جستجوی عمق نخست بحث کردیم و دو کاربرد آن در پیدا کردن مولفه‌های همبندی و همچنین پیدا کردن راس‌های برشی را مورد بررسی قرار دادیم. فرم کلی جستجوی عمق نخست به صورت زیر است:

```
DFS( i ){
        mark[i] = true
        for every j in N(i)
                if(!mark[j])
                        DFS(j)
}
```

حال زمان اجرای الگوریتم DFS را بررسی می‌کنیم. در پیاده‌سازی با استفاده از ماتریس مجاورت، از آنجا که n راس داریم و به دست آوردن همسایه‌های هر راس از $O(n)$ می‌باشد؛ پس زمان کلی از مرتبه $O(n^2)$ است. با استفاده از لیست مجاورت، به دست آوردن همسایه‌های هر راس i از مرتبه $O(d_i)$ می‌باشد؛ پس مرتبه کلی الگوریتم از $O(\sum_i d_i) = O(m)$ می‌باشد.(n تعداد رأس‌ها و m تعداد یال‌ها در گراف اصلی است.)

درخت DFS: درختی است که راس‌های آن، راس‌های گراف اصلی و یال‌های آن، یال‌هایی از گراف اصلی هستند که از طریق آن‌ها DFS انجام شده است. در شکل زیر درخت DFS مربوط به گراف سمت چپ را مشاهده می‌کنیم:



سپس به ازای هر راس دو پارامتر $discover$ و $finish$ را معرفی کردیم:

- $discovertime$: زمانی که DFS یک راس آغاز می‌شود.
- $finishtime$: زمانی که DFS یک راس پایان می‌یابد.

همچنین دو نوع یال $backward$ و $cross$ را مورد بررسی قرار دادیم:

- یال $backward$: اگر i یا از اجداد j باشد یا برعکس، یال (i,j) یال $backward$ است.
- یال $cross$: اگر نه i از اجداد j باشد و نه j از اجداد i، یال (i,j) یال $cross$ می‌باشد.

مشاهده کردیم که اگر یال (i,j) در گراف اصلی وجود داشته باشد، آنگاه در درخت DFS یا i از اجداد j است یا j از اجداد i.

حال کاربردهای جستجوی عمق نخست را بررسی می‌کنیم:

۱. **یافتن تعداد مولفه همبندی:** برای به‌دست آوردن تعداد مولفه همبندی می‌توان از قطعه کد زیر استفاده کرد:

```
k = 0
for every 1 ≤ i ≤ n
        if(!mark[i])
                k ++
                DFS( i )
```

بنابراین زمان اجرا آن با استفاده از ماتریس مجاورت $O(n^2)$ و با استفاده از لیست مجاورت از مرتبه $O(m+n)$ می‌باشد.

۲. **پیدا کردن راس‌های برشی:**

راه حلّ ۱:

۱) ابتدا تعداد مولفه‌های همبندی را پیدا کنیم. $\Leftarrow O(n+m)$

۲) سپس به ازای هر راس i، i را حذف کرده و تعداد مولفه‌های همبندی در گراف جدید را به دست می‌آوریم. $\Leftarrow$ این مرحله n بار با زمان $O(n+m)$ انجام می‌شود پس به طور کلی از $O(n^2+nm)$ می‌باشد.

در نهایت اگر تعداد مولفه‌های همبندی اولیه‌ی گراف با تعداد مولفه‌های همبندی پس از حذف راس متفاوت باشد، آن راس برشی است.

راه حلّ ۲:

۱) درخت DFS را پیدا می‌کنیم.

۲) به ازای هر راس، بالاترین جدی که به آن یال دارد را می‌یابیم. $\Leftarrow O(m)$

۳) به ازای هر راس، بالاترین جدی که زیردرختان فرزندان آن راس به آن متصل هستند را پیدا می‌کنیم.

۴) در نهایت یک رأس برشی **نیست**، اگر تمام زیردرختان فرزندانش به یکی از رأس‌های بالایی وصل باشند.

تمام مراحل گفته شده را می‌توان با یک DFS انجام داد. بنابراین کلّ زمان این الگوریتم تنها مربوط به DFS می‌شود که آن نیز در بهترین حالت زمان $O(n+m)$ دارد.