



دانشکده مهندسی کامپیوتر

جزوه درس

ساختمان‌های داده

استاد درس: سید صالح اعتمادی

پاییز ۱۳۹۸

## جلسه ۱۸

# Disjoint Sets

امید میرزاجانی - ۱۳۹۸/۸/۲۰

جزوه جلسه ۱۸ ام مورخ ۱۳۹۸/۸/۲۰ درس ساختمان‌های داده تهیه شده توسط امید میرزاجانی. این ساختمان داده، کاربرد های بسیاری دارد که یک از آنها تشخیص دادن هم گروه بودن تو راس در یک گراف است. به طور مثال وقتی میخواهیم ببینیم آیا از راس  $a$  به راس  $b$  مسیر هست یا خیر، میتوانیم از این ساختمان داده استفاده کنیم.

## ۱.۱۸ عملیات ساپورت شده Disjoint Sets

اعمال ویژه این ساختمان داده به شرح زیر است:

- $\text{MakeSet}(x)$ : که عنصر  $x$  را به یک مجموعه خاص مرتبط میکند. در واقع به آن عنصر یک ID ویژه میدهد.
- $\text{Find}(x)$ : ID آن عنصر را به عنوان خروجی میدهد. از این متد میتوان فهمید که آیا دو عنصر  $a, b$

داخل یک مجموعه قرار میگیرند یا خیر.

**Data:** Two elements (a , b)

**Result:** Check are these elements are in one set or not

initialization;

**if**  $Find(a) == Find(b)$  **then**

    | Return "yes";

**else**

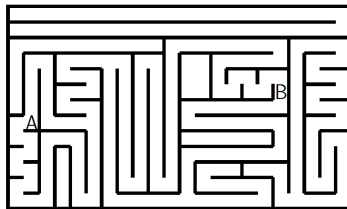
    | Return "no";

**end**

- $Union(x,y)$ : این متد دو مجموعه را با هم ادغام میکند. در واقع بعد از اجرا شدن این متد، همه عناصری که با x هم گروه بودند؛ با تمام عناصری که هم گروه y اند، به یک مجموعه تبدیل میشوند.

## ۲.۱۸ یک مثال از Disjoint Sets

فرض کنید میخواهیم پیدا کنیم آیا از A میتوان به B رفت یا خیر؟



شکل ۱.۱۸: یک مثال معروف!

ابتدا برای هر خانه از این شکل متد Makeset را فراخوانی میکنیم تا ID منحصر به فرد آن شکل بگیرد. سپس به ازای هر خانه با استفاده از متد Find چک میکنیم که هر خانه با کدام ۴ خانه مجاورش در یک گروه قرار دارد. سپس همانطور که در بالا توضیح داده شد، چک میکنیم که آیا دو عنصر A ، B در یک مجموعه قرار گرفته اند یا خیر.

## ۳.۱۸ پیاده سازی

این ساختمان داده را میتوان به صورت آرایه<sup>۱</sup> یا لینکدلیست<sup>۲</sup> پیاده کرد.

### ۱.۳.۱۸ با آرایه

همانطور که واضح است برای پیاده سازی این ساختمان داده ابتدا باید سه متو ذکر شده در بالا را پیاده سازی کرد. برای پیاده سازی ابتدا یک آرایه در نظر میگیریم که هر خانه اش شامل ID آن عنصر است. هم چنین ID هر عنصر، مقدار کوچکترین عضو آن مجموعه است. اگر این آرایه را `smallest` بنامیم داریم:

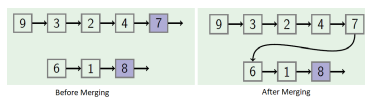
`Makeset(i)` : اندیس شماره `i` آرایه `smallest` را برابر `i` قرار میدهد. (این متد با  $O(1)$  قابل بررسی است.)

`Find(i)` : `smallest[i]` را بر میگرداند. (این متد با  $O(1)$  قابل بررسی است.)

`Union(i,j)` : بین ID های `i` ، `j` کوچکترین ID را `minID` مینامیم. سپس در آرایه `smallest` پیمایش میکنیم و هر خانه ای که برابر `Find(i)` یا `Find(j)` بود، برابر `minID` قرار میدیم. (این متد با  $O(n)$  قابل بررسی است، زیرا با یک حلقه تمامی اعضای `smallest` را پیمایش کردیم و در هر بار یک عمل `Find`) با اردر ۱ انجام دادیم.

### ۲.۳.۱۸ با لینکدلیست

در این نوع از پیاده سازی هر عضو را داخل یک لینکد لیست میگذاریم. و همچنین برای متد `Find` آخرین عضو آن لینکد لیست را در نظر میگیریم. همچنین برای ادغام دو مجموعه که در اینجا همان ادغام کردن دو لینکدلیست است، اشاره گر آخرین عضو مجموعه اول را برابر اولین عضو مجموعه دوم قرار میدهیم.

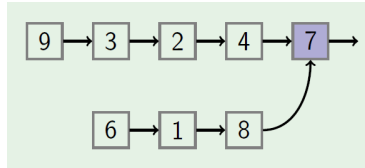


شکل ۲.۱۸: ادغام سازی

<sup>۱</sup>Array  
<sup>۲</sup>LinkedList

## ۳.۳.۱۸ نوع دیگری از لینکدلیست

این روش با روش قبل شباهت بسیاری دارد با این تفاوت که برای ادغام کردن، مانند شکل عمل میکنیم.



شکل ۳.۱۸: ادغام سازی

اینگونه در نهایت یک شکلی مانند گراف جهت دار و بدون دور داریم و میتوان آن را به درخت تشبیه کرد که ID هر عضو ریشه ی درخت مرتبط است. این نوع پیاده سازی متد Union را بهبود میدهد ولی برای متد Find با  $O(n)$  انجام میدهد زیرا برای رسیدن به ID مورد نظر باید تا آخر لینکدلیست مورد نظر پیمایش کنیم. ابتدا یک آرایه parent در نظر میگیریم که در آن هر عضو آرایه برابر پدر آن عضو است. Makeset(i) : اندیس شماره i آرایه parent را برابر i قرار میدهد. (این متد با  $O(1)$  قابل بررسی است.)  
: Find(i)

```

while  $i \neq \text{parent}(i)$  do
  |  $i = \text{parent}(i)$ ;
end
return i;
  
```

(این متد با  $O(\text{tree height})$  قابل بررسی است.)

Union : برای ادغام دو مجموعه ریشه مجموعه اول را فرزند ریشه ی مجموعه دوم قرار میدهیم. اما باید توجه داشته باشیم که برای یک درخت، هر چه ارتفاع آن کمتر باشد به برنامه ما بهبود میبخشد زیرا هر اطلاعاتی که بخواهیم از درخت بگیریم در زمان کمتری انجام میشود. پس درختی را فرزند دیگری قرار میدهیم که در نتیجه ی کار ارتفاع درخت نهایی کمتر شود.