图应用

Kruskal算法：并查集

11-F2

Following the leader, the leader, the leader,

We're following the leader wherever he may go.

这里的人事关系是由一个个“单位”组成的......白天里“单位”是魂，
人活在一个一个的单位里......我很庆幸，我是个有单位的人

邓俊辉
deng@tsinghua.edu.cn

Union-Find

- ❖ 给定一组互不相交的等价类

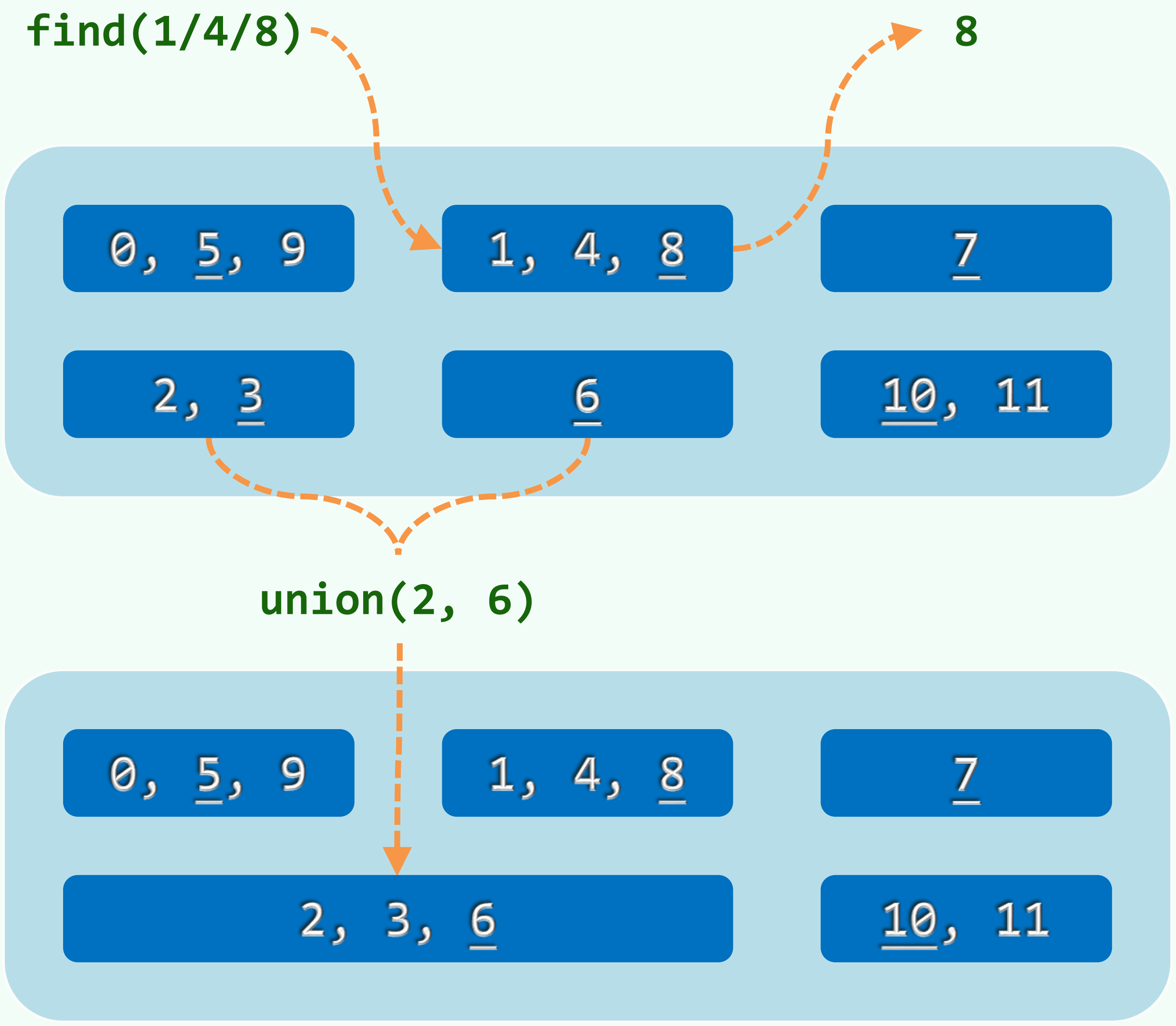
  各由其中一个成员作为代表

- ❖ Find(x)：找到元素x所属等价类

- ❖ Union(x, y)：合并x和y所属等价类

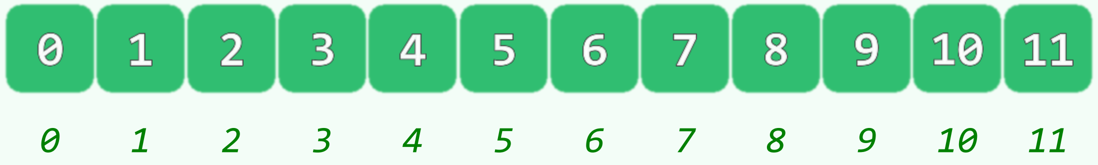
- ❖ Singleton：初始时各包含一个元素

- ❖ Kruskal = Union-Find

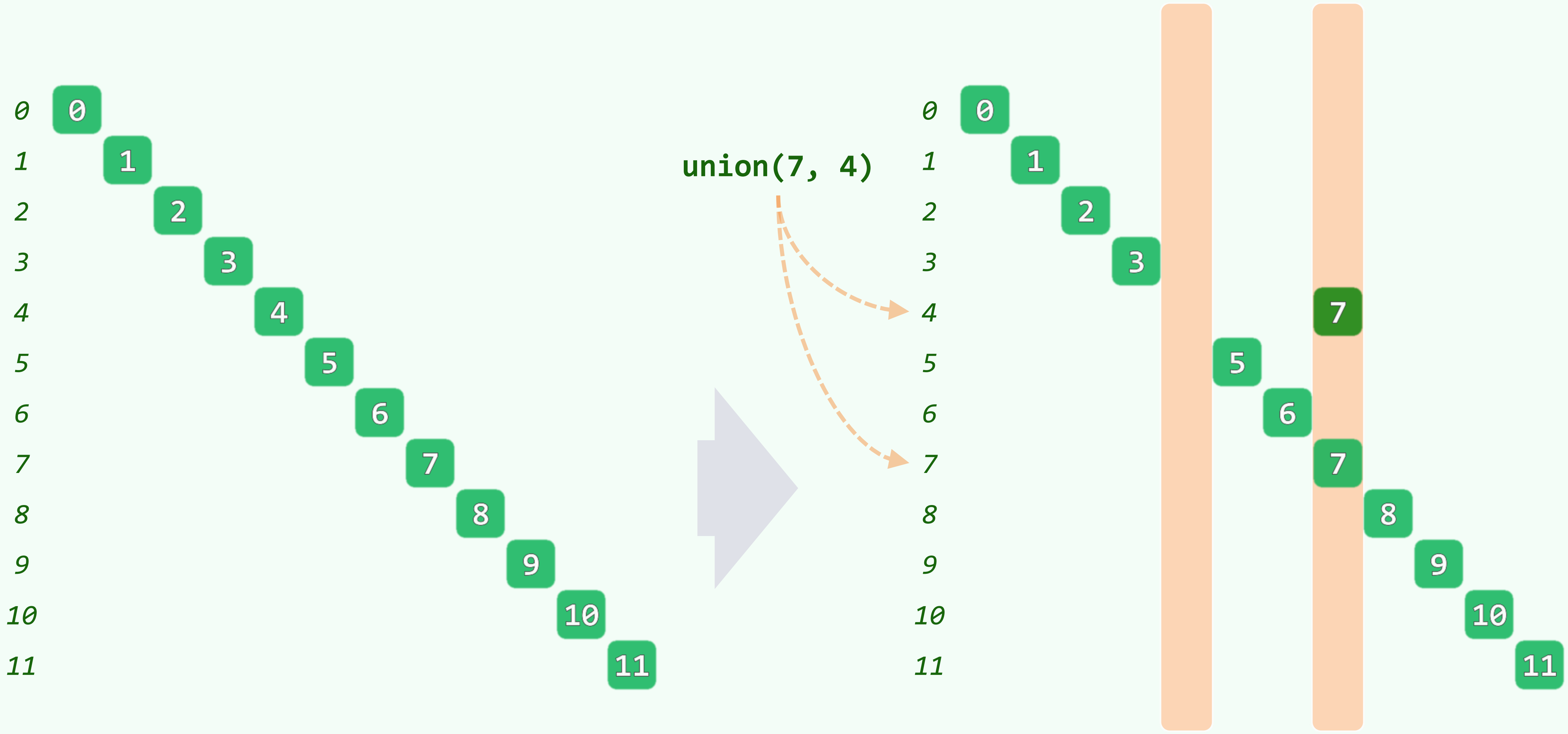


Quick-Find

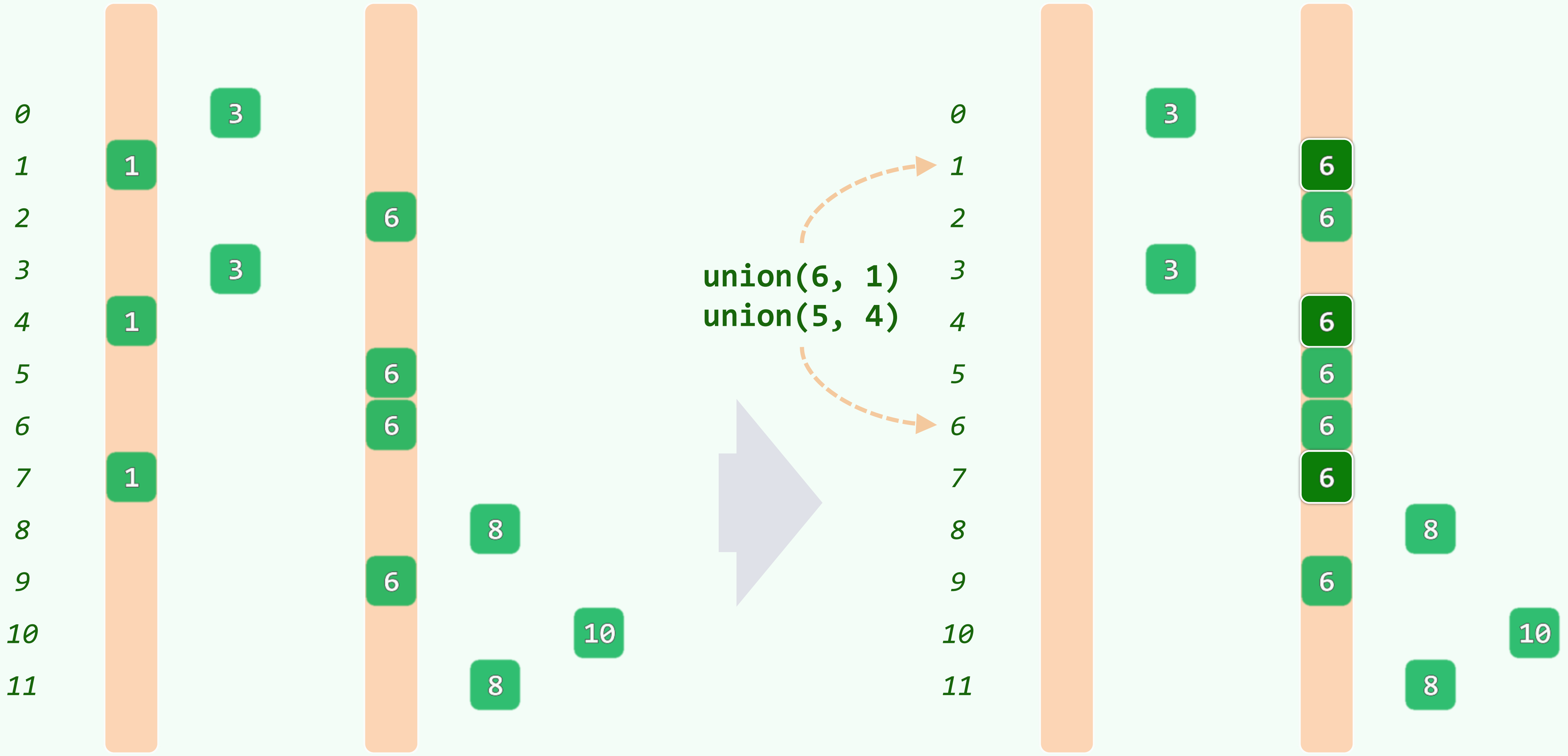
```
class UnionFind:

    def __init__(self, n): #group[]记录各元素所属子集；初始各成一类，以[0,n)间整数标识

        self.g = self.n = n; self.group = [ k for k in range(n) ]

    def find(self, k):

        return self.group[k]

    def union(self, i, j):

        iGroup , jGroup = self.group[i] , self.group[j]

        if iGroup == jGroup: return

        for k in range(self.n):

            if (self.group[k] == jGroup): self.group[k] = iGroup

        self.g -= 1
```



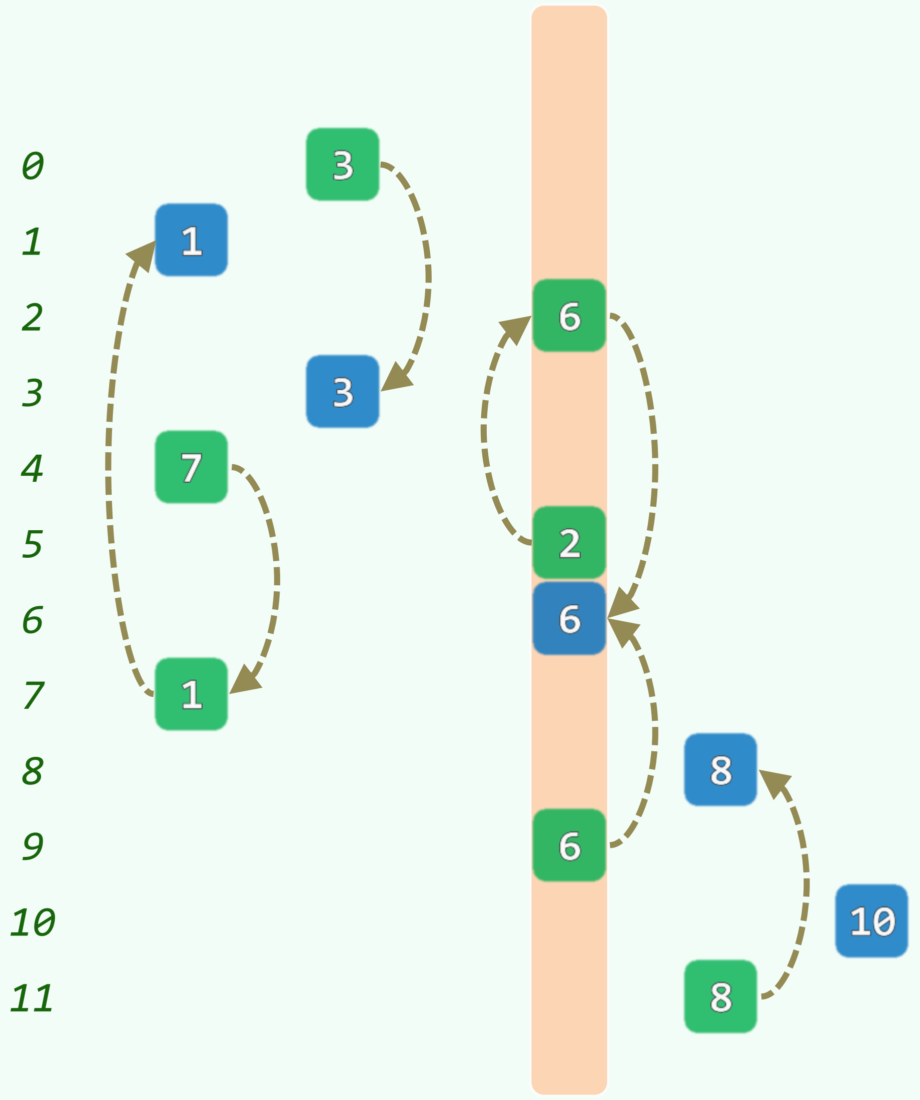
Slow-Union (1/2)



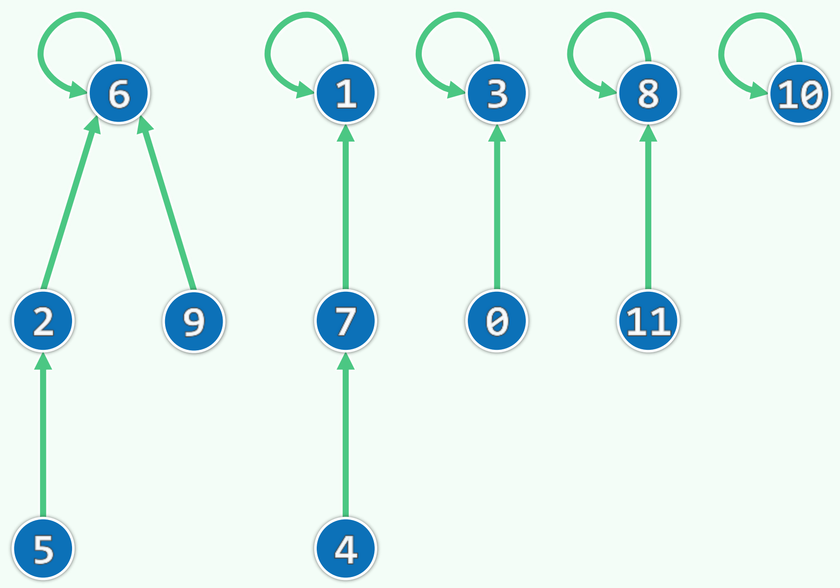
Slow-Union (2/2)



Group ~ Parent ~ Root

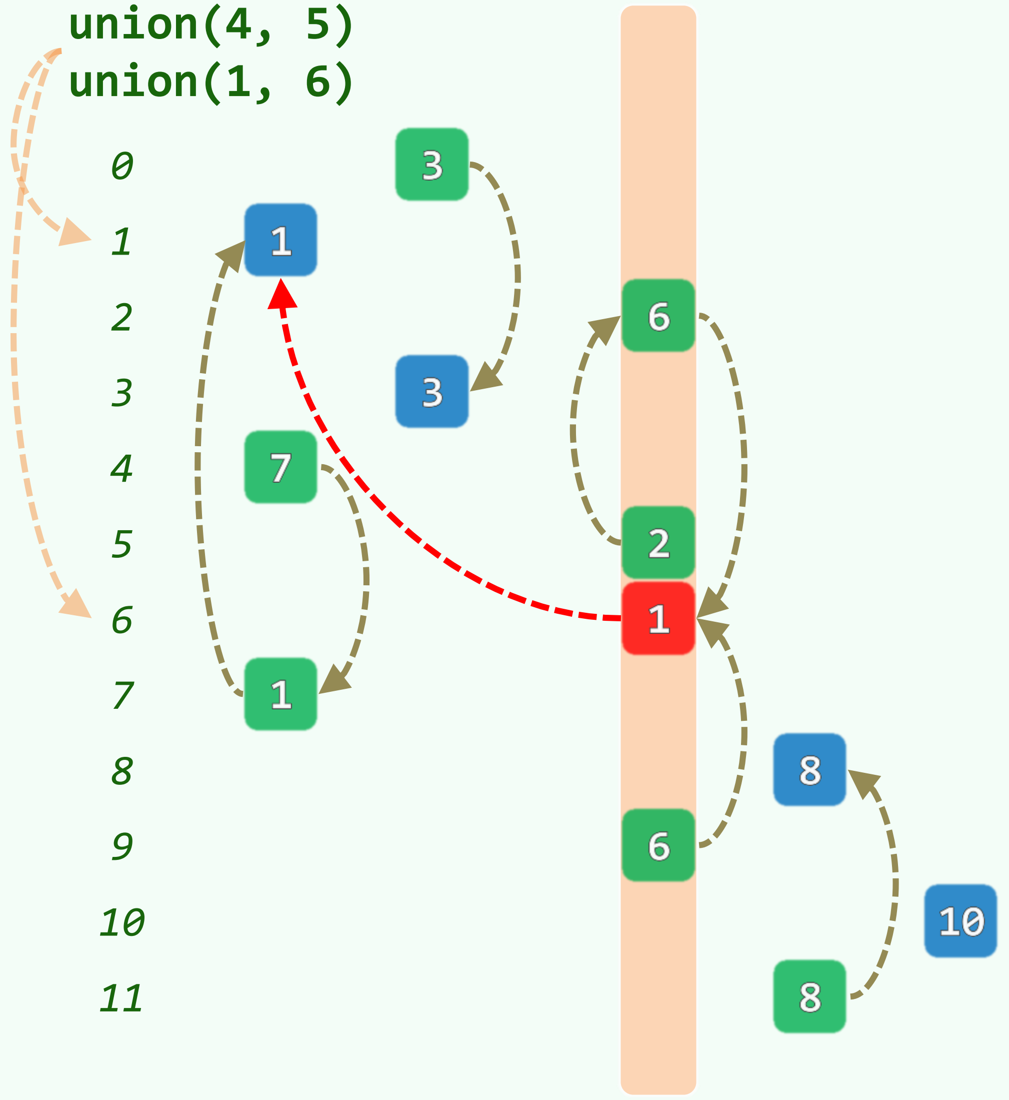


rank	parent
0	3
1	1
2	6
3	3
4	7
5	2
6	6
7	1
8	8
9	6
10	10
11	8

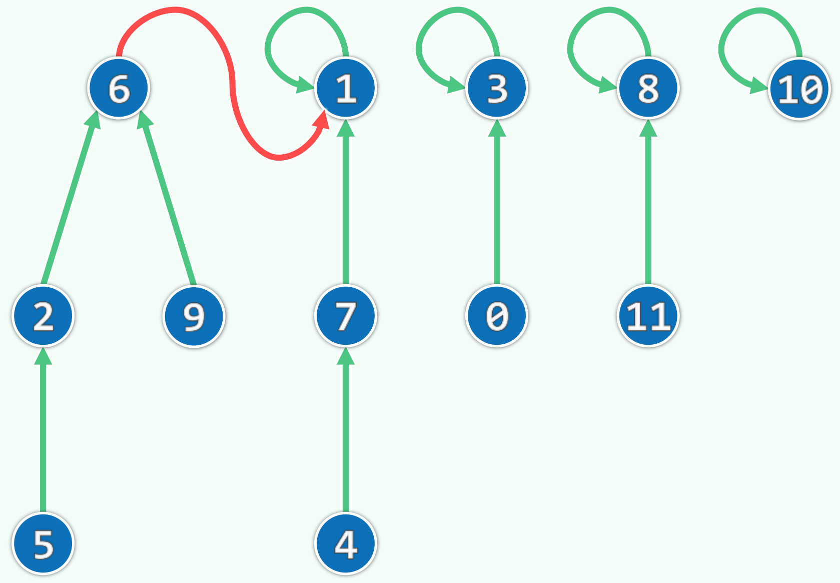


Quick-Union

```
union(4, 5)
union(1, 6)
```

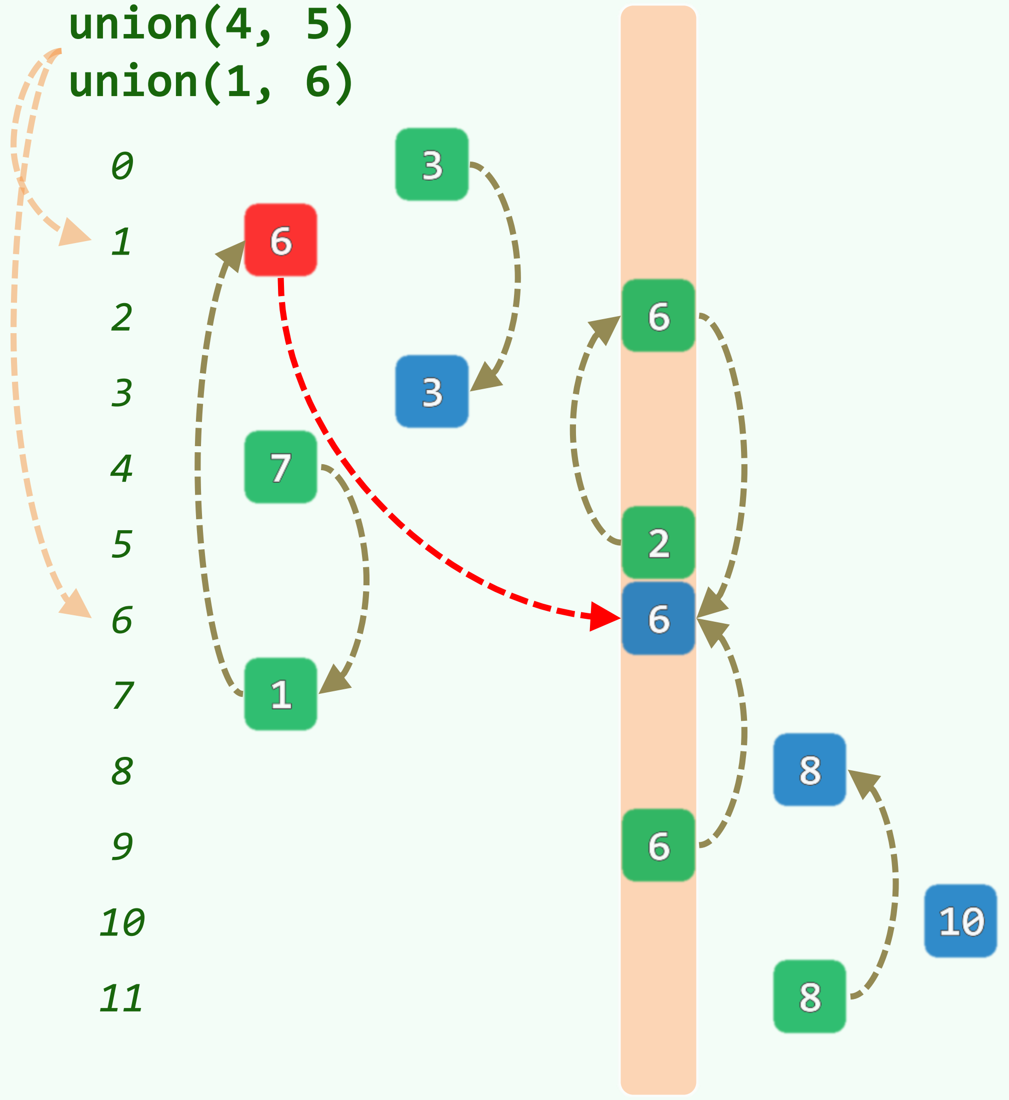


rank	parent
0	3
1	1
2	6
3	3
4	7
5	2
6	1
7	1
8	8
9	6
10	10
11	8

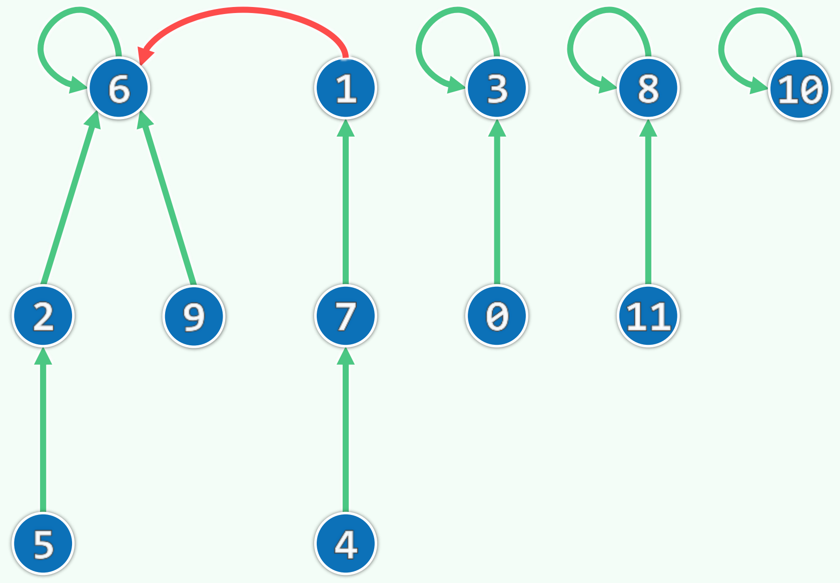


Weighting By Size Or Height

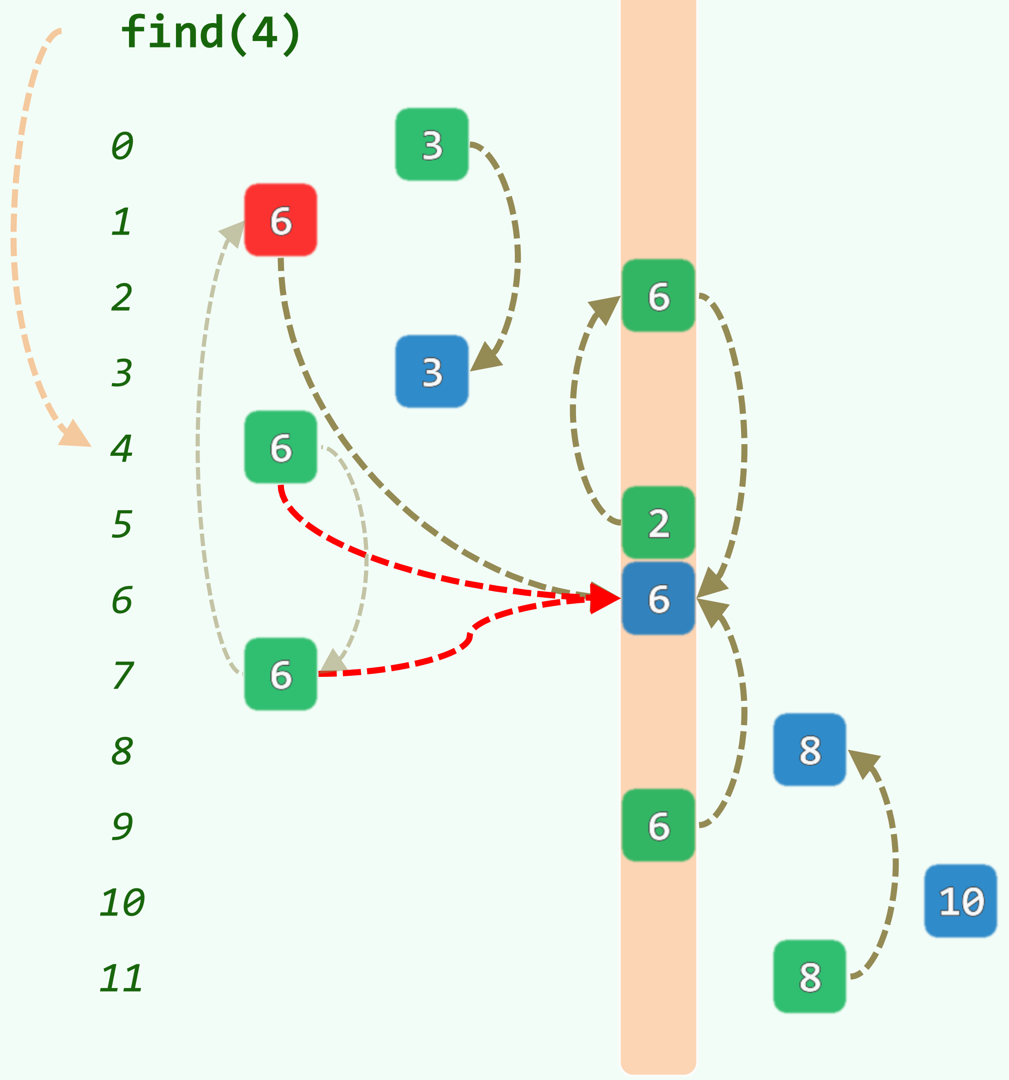
```
union(4, 5)
union(1, 6)
```



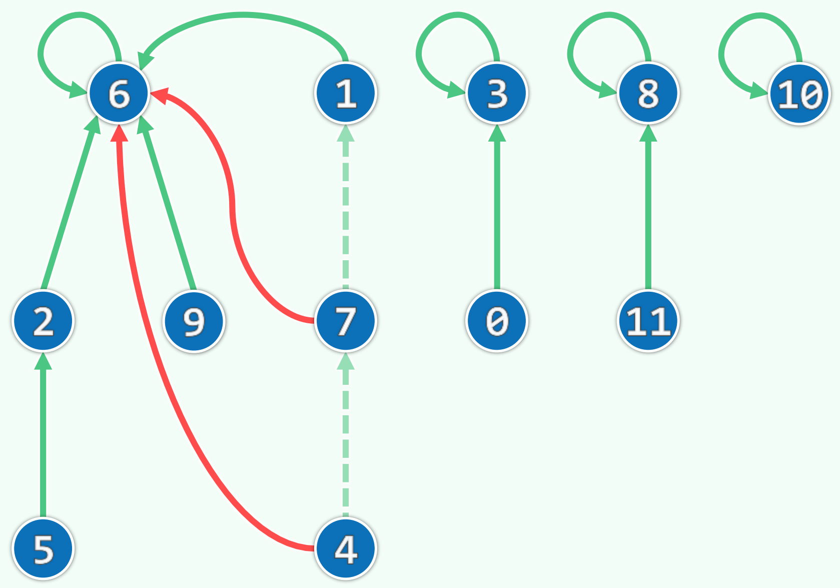
rank	parent	size
0	3	
1	6	~~3~~
2	6	
3	3	2
4	7	
5	2	
6	6	~~4~~7
7	1	
8	8	2
9	6	
10	10	1
11	8	



Path Compression/Folding



rank	parent	size
0	3	
1	6	
2	6	
3	3	2
4	6	
5	2	
6	6	7
7	6	
8	8	2
9	6	
10	10	1
11	8	



6
1
3
8
10
2
9
7
0
11
5
4