



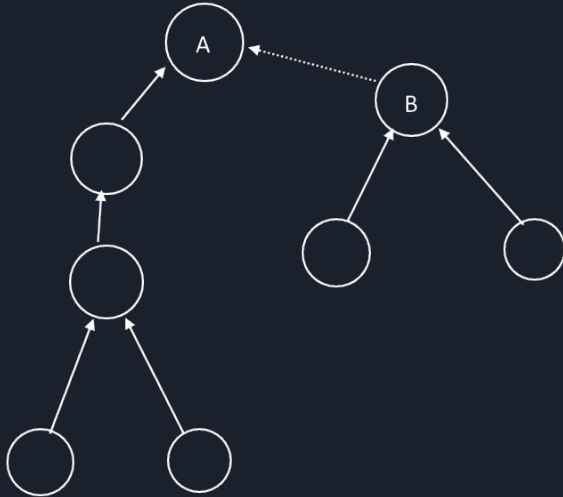
# ProgTeam Week 8

Union Find

# Union Find

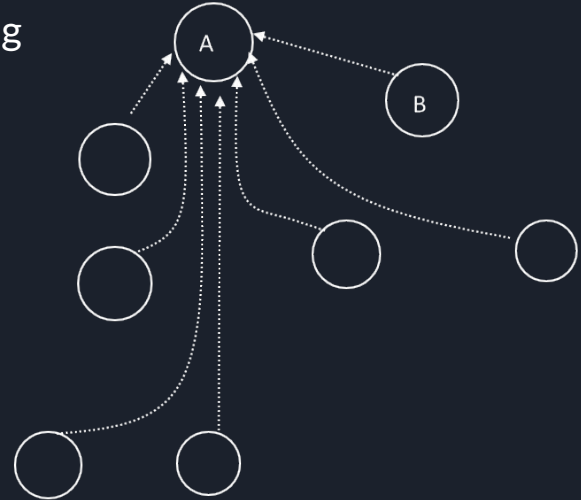
Disjoint-Set Data Structure

- Sometimes we want to know if two elements are of the same set
- Copying entire set can be expensive
- Solution: have pointers to parents when we combine two sets



# Union Find

- Sometimes we want to know if two elements are of the same set
- Copying entire set can be expensive
- Solution: have pointers to parents when we combine two sets
- New Problem: paths to root can get very long
- Solution: path compression





# Find:

```
int root[N];  
int find(int x){  
    if(root[x] == -1) return x;  
    return root[x] = find(root[x]);  
}
```



## Union:

```
void join(int x, int y){  
    x = find(x);  
    y = find(y);  
    if(x == y) return;  
    root[x] = y;  
    // if(root[x] > root[y]) swap(x, y);  
    // root[x] += root[y];  
    // The above is a _little_ faster in theory but usually not  
    noticeable  
}
```