

알고리즘 멘토링

- Union Find -

- 주민찬 -

Union-Find

Disjoint-Set 알고리즘이라고도 불리는 알고리즘으로 구체적으로 여러 개의 노드가 존재할 때 두 개의 노드를 선택해서, 현재 이 두 노드가 서로 같은 그래프에 속하는지 판별하는 알고리즘입니다.

Parent

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

Rank

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

Parent의 각 인덱스는 노드의 번호

인덱스의 값에는 그 노드의 parent node가 적혀 있음 루트일 경우 자기 자신이 적혀 있음

Rank의 각 인덱스는 노드의 번호

인덱스의 값에는 그 노드가 속한 트리의 노드의 수가 적혀 있음

Union-Find

Union-Find 에게는 두 가지의 연산이 있다.
노드의 루트 노드를 찾는 Find()
두 트리를 합치는 Union()

```
def find(x):  
    if x != parent[x]:  
        parent[x] = find(parent[x])  
  
    return parent[x]
```

연산 횟수를 줄이기 위해 parent를 계속 변경

```
def union(x,y):  
    x_root = find(x)  
    y_root = find(y)  
  
    if x_root != y_root:  
        if rank[x_root] < rank[y_root]:  
            parent[x_root] = y_root  
        elif rank[x_root] > rank[y_root]:  
            parent[y_root] = x_root  
        else:  
            parent[x_root] = y_root  
            rank[x_root] += rank[y_root]  
            rank[y_root] = rank[x_root]
```

루트를 찾아 두 트리의 Rank를 비교
작은 쪽에다가 붙임

Union-Find

Parent

index	0	1	2	3	4	5	6	7	8	9	10
parent	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

또 다른 표현식

Parent의 각 인덱스는 노드의 번호

인덱스의 값에는 그 노드의 parent node가 적혀 있음

루트일 경우 그 트리의 노드의 수가 음수로 적혀 있음

Union-Find

```
def find(x):  
    if (parent[x] < 0): return x  
    parent[x] = find(parent[x])  
    return parent[x]
```


연산 횟수를 줄이기 위해 parent를 계속 변경

```
def union(x,y):  
    x_root = find(x)  
    y_root = find(y)  
  
    if x_root != y_root:  
        total = parent[x_root] + parent[y_root]  
        if parent[x_root] < parent[y_root]:  
            parent[y_root] = x_root  
            parent[x_root] = total  
        elif parent[x_root] > parent[y_root]:  
            parent[x_root] = y_root  
            parent[y_root] = total  
        else:  
            parent[x_root] = y_root  
            parent[y_root] = total
```

루트를 찾아 두 트리의 노드의 수를 비교
적은 쪽에다가 붙임

Union-Find

- 시간복잡도: $O(\alpha(N)) \approx O(1)$ $\alpha(N)$: 애커만 함수의 역함수
 - $1 \leq N < 3$: $\alpha(N) = 1$
 - $3 \leq N < 7$: $\alpha(N) = 2$
 - $7 \leq N < 63$: $\alpha(N) = 3$
 - $63 \leq N < 2^{2^{2^{2^{2^2}}}}$: $\alpha(N) = 4$

A lifebuoy with orange and white segments is floating in dark, rippling water. A large splash of water is rising from the center of the lifebuoy, creating a vertical column of water droplets. The background is a dark, hazy sky or water surface.

실습!!