

알고리즘 중급 세미나

05: 분리 집합

연세대학교 전우제^{kiwiyou}

2023.01.15.r1

분리 집합

- 정점 V 개가 있을 때, 다음 쿼리 Q 개를 처리하기
 - 1 $u \ v$: u 와 v 를 잇는 양방향 간선을 추가
 - 2 $u \ v$: u 와 v 사이를 이동할 수 있는지 출력
- DFS를 이용하면 1번 쿼리에 $\mathcal{O}(1)$, 2번 쿼리에 $\mathcal{O}(V)$
- 더 빠르게 할 수 있을까?

분리 집합

- 집합 N 개에 각각 대표 원소 $1, 2, \dots, N$ 이 있을 때, 다음 연산을 처리
 - 원소 a 가 포함된 집합의 대표 원소 찾기: `find(a)`
 - 집합 A 와 B 를 모두 두 집합의 합집합으로 대체: `union(a, b)`
- A 가 B 를 포함하면 B 의 부모를 A 로 설정, 전체 집합을 트리 구조로 표현
- `find(a)`는 트리의 루트를 찾는 연산
- `union(a, b)`는 루트와 루트 사이에 간선을 추가하는 연산

분리 집합

- find의 최적화
- 편의상 집합 번호 = 대표 원소로 두고, 부모가 자신과 같으면 루트로 정의

```
1: function FIND( $a$ )  
2:    $r \leftarrow a$   
3:   while parent( $r$ )  $\neq r$  do  
4:      $r \leftarrow$  parent( $r$ )  
5:   return  $r$ 
```

- 매번 루트까지 올라가야 하므로 최악 $\mathcal{O}(N)$

분리 집합

- 루트까지 올라간 결과를 저장해 두기

```
1: function FIND( $a$ )  
2:    $r \leftarrow a$   
3:   while parent( $r$ )  $\neq r$  do  
4:      $r \leftarrow \text{parent}(r)$   
5:   parent( $a$ )  $\leftarrow r$   
6:   return  $r$ 
```

- 여전히 amortized $\mathcal{O}(N)$
- 재귀를 이용하여 매 반복마다 저장해 둔다면?

분리 집합

- 경로 압축 Path Compression: amortized $\mathcal{O}(\log N)$ - 증명 생략

```
1: function FIND( $a$ )
2:   if parent( $a$ )  $\neq a$  then
3:     parent( $a$ )  $\leftarrow$  find(parent( $a$ ))
4:   return parent( $a$ )
```

```
1: function FIND( $a$ )
2:    $r \leftarrow a$ 
3:   while parent( $r$ )  $\neq r$  do
4:     parent( $r$ )  $\leftarrow$  parent(parent( $r$ ))
5:      $r \leftarrow$  parent( $r$ )
6:   return  $r$ 
```

분리 집합

- union의 최적화
- Union by size

```
1: function UNION( $a, b$ )  
2:   if size( $a$ ) < size( $b$ ) then  
3:     UNION( $b, a$ )  
4:   size( $a$ )  $\leftarrow$  size( $a$ ) + size( $b$ )  
5:   parent( $b$ )  $\leftarrow a$ 
```

- 경로 압축 없이 amortized $\mathcal{O}(\log N)$, 경로 압축과 함께 amortized $\mathcal{O}(\alpha(N))$

분리 집합

- Union by rank

```
1: function UNION( $a, b$ )  
2:   if rank( $a$ ) < rank( $b$ ) then  
3:     UNION( $b, a$ )  
4:   if rank( $a$ ) = rank( $b$ ) then  
5:     rank( $a$ )  $\leftarrow$  rank( $a$ ) + 1  
6:   parent( $b$ )  $\leftarrow a$ 
```

- 경로 압축 없이 amortized $\mathcal{O}(\log N)$, 경로 압축과 함께 amortized $\mathcal{O}(\alpha(N))$
- rank의 값은 최대 $\log N$

분리 집합

- 실시간으로 사이클 판정 가능
- 연결 요소 개수를 실시간으로 추적 가능
- 다음 빈 공간 찾기

과제

- [11085 군사 이동](#)
- [14595 동방 프로젝트 \(Large\)](#)
- [29618 미술 시간](#)

최소 스패닝 트리

- 스패닝 트리: 양방향 연결 그래프의 모든 정점을 포함하는 트리
- 최소 스패닝 트리: 스패닝 트리 중 트리 간선의 가중치 합이 최소인 것
- Prim, Kruskal 외 여러 가지 알고리즘이 있음

Prim 알고리즘

- 한 정점에서 시작해서 가중치가 가장 작은 간선을 계속 붙여 나감
- 붙일 수 있는 간선의 목록을 우선순위 큐로 관리하면 $\mathcal{O}(E \log V)$

Kruskal 알고리즘

- 가중치가 작은 간선부터 차례대로, 사이클을 이루지 않는다면 트리에 추가
- 정렬에 $\mathcal{O}(E \log E)$, 사이클 확인에 $\mathcal{O}(E\alpha(V))$

과제

- 27945 슬슬 가지를 먹지 않으면 죽는다
- 28473 도로 위의 표지판