노트북: 첫 번째 노트북
만든 날짜: 2020-11-23 오후 11:45
URL: https://continuous-development.tistory.com/198

Algorithm

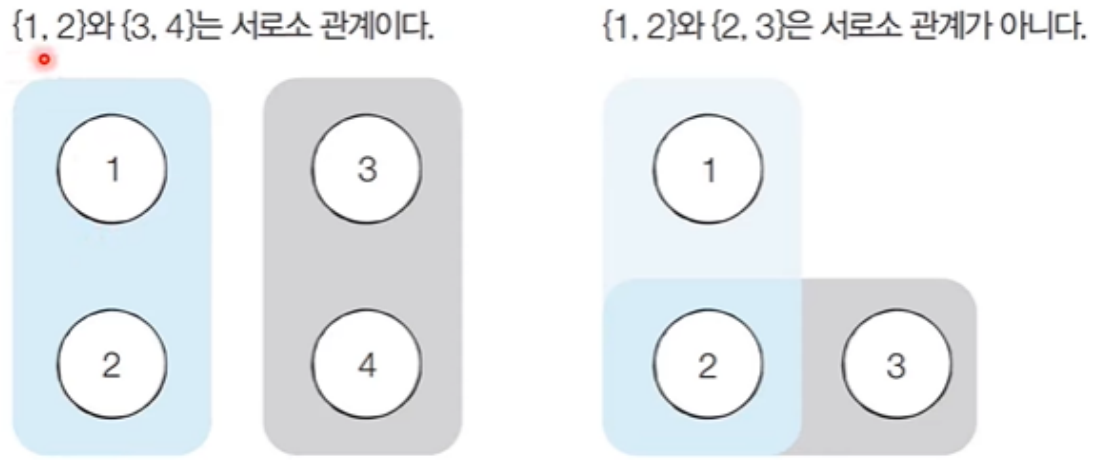
[Algorithm] 33강 : 서로소 집합 자료구조의 정의와 구현

2020. 11. 23. 23:44 수정 삭제 공개

서로소 집합

1.1 서로소 집합이란?

- **서로소 집합(Disjoint Sets)란 공통 원소가 없는 두 집합을 의미한다.**



1.2 서로소 집합 자료구조란?

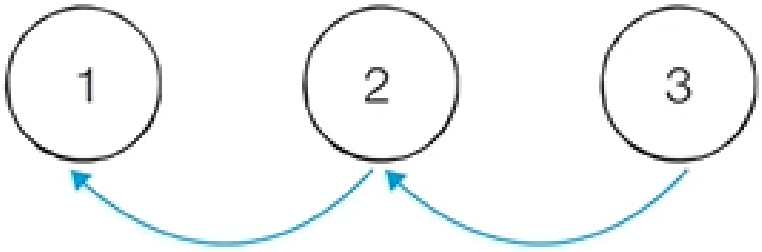
- 서로소 부분 집합들로 나누어진 원소들의 데이터를 처리하기 위한 자료구조이다.
- 서로소 집합 자료구조는 두 종류의 연산을 지원한다.
  - **합집합(Union)** : 두 개의 원소가 포함된 집합을 하나의 집합으로 합치는 연산
  - **찾기(Find)** : 특정한 원소가 속한 집합이 어떤 집합인지 알려주는 연산
- 서로소 집합 자료구조는 **합치기 찾기(Union) 자료구조**라고 불려진다.

1.3 서로소 집합 자료구조 동작 과정

1. 합집합 연산을 확인하여, 서로 연결된 두 노드 A, B를 확인한다.
   1. A와 B의 루트 노드 A`, B`를 각각 찾는다.
   2. A`를 B`의 부모 노드로 설정한다
2. 모든 합집합 연산을 처리할 때까지 1번의 과정을 반복한다.

1.4 서로소 집합 자료구조:연결성

- 기본적인 형태의 서로소 집합 자료구조에서는 루트 노드에 즉시 접근할 수 없다
  - 루트 노드를 찾기 위해 부모 테이블을 계속해서 확인하며 거슬러 올라가야 한다
- 다음 예시에서 노드 3의 루트를 찾기 위해서는 노드 2를 거쳐 노드 1에 접근해야 한다.



노드 번호	1	2	3
부모	1	1	2

1.5 기본적인 구현 방법

```
# 특정 원소가 속한 집합을 찾기
def find_parent(parent,x):
    #루트 노드를 찾을 때까지 재귀 호출
    if parent[x] != x:
        return find_parent(parent,parent[x])
    retrun x

# 두 원소가 속한 집합을 합치기
def union_parent(parent,a,b):
    a = find_parent(parent,a)
    b = find_parent(parent,b)
    if a < b:
        parent[b] = a
    else:
        parent[a] = b

# 노드의 개수와 간선의 개수 입력받기
v, e = map(int,input().split())
parent = [0] * (v+1) # 부모 테이블 초기화 하기

# 부모 테이블상에서, 부모를 자기 자신으로 초기화
for i in range(1,v+1):
    parent[i] = i

# union 연산을 각각 수행
for i in range(e):
    a, b = map(int,input().split())
    union_parent(parent, a, b)

# 각 원소가 속한 집합 출력하기
print('각 원소가 속한 집합:' , end='')
```

```
for i in range(1, v+1):
    print(find_parent(parent,i),end= ' ')
print()

# 부모 테이블 내용 출력하기
print('부모 테이블: ', end ='')
for i in range(1, v+1):
    print(parent[i], end='')
```

1.6 기본적인 구현 방법의 문제점

- **합집합 연산이 편향되게 이루어지는 경우 찾기 함수가 비효율적으로 동작한다**
- **최악의 경우에는 찾기 함수가 모든 노드를 다 확인하게 되어 시간 복잡도가 O(V)이다**
    - **다음과 같이 {1,2,3,4,5}의 총 5개의 원소가 존재하는 상황 일 때**
- **수행된 연산들**: $Union(4,5)$, $Union(3,4)$, $Union(2,3)$, $Union(1,2)$

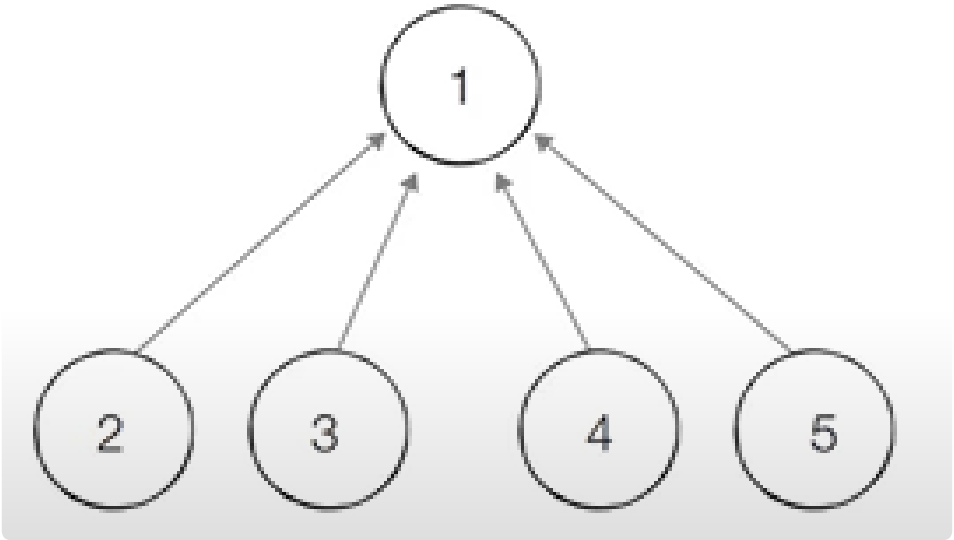
노드 번호	1	2	3	4	5
부모	1	1	2	3	4

1.7 경로 구축

- **찾기 함수를 최적화하기 위한 방법으로 경로 압축을 이용할 수 있다.**
    - 찾기 함수를 재귀적으로 호출한 뒤에 부모 테이블 값을 바로 경신한다.

```python
# 특정 원소가 속한 집합을 찾기
def find_parent(parent, x):
    # 루트 노드가 아니라면, 루트 노드를 찾을 때까지 재귀적으로 호출
    if parent[x] != x:
        parent[x] = find_parent(parent, parent[x])
    return parent[x]
```

- 경로 압축 기법을 적용하면 각 노드에 대하여 찾기 함수를 호출한 이후에 해당 노드의 루트 노드가 바로 부모 누드가 된다
- 동일한 예시에 대해서 모든 합집합 함수를 처리한 후 각 원소에 대하여 찾기 함수를 수행하면 다음과 같이 부모 테이블이 갱신된다
- 기본적인 방법에 비하여 시간 복잡도가 개선된다.



1.8 경로구축 구현

```python
# 특정 원소가 속한 집합을 찾기
def find_parent(parent,x):
    #루트 노드를 찾을 때까지 재귀 호출
    if parent[x] != x:
        parent[x] = find_parent(parent, parent[x])
    retrun parent[x]

# 두 원소가 속한 집합을 합치기
def union_parent(parent,a,b):
    a = find_parent(parent,a)
```

```
    b = find_parent(parent,b)
    if a < b:
        parent[b] = a
    else:
        parent[a] = b

# 노드의 개수와 간선의 개수 입력받기
v, e = map(int,input().split())
parent = [0] * (v+1) # 부모 테이블 초기화 하기

# 부모 테이블상에서, 부모를 자기 자신으로 초기화
for i in range(1,v+1):
    parent[i] = i

# union 연산을 각각 수행
for i in range(e):
    a, b = map(int,input().split())
    union_parent(parent, a, b)

# 각 원소가 속한 집합 출력하기
print('각 원소가 속한 집합:' , end='')
for i in range(1, v+1):
    print(find_parent(parent,i),end= ' ')
print()

# 부모 테이블 내용 출력하기
print('부모 테이블: ', end ='')
for i in range(1, v+1):
    print(parent[i], end='')
```


'Algorithm' 카테고리의 다른 글

[Algorithm] 33강 : 서로소 집합 자료구조의 정의와 구현

[Algorithm] 33강 : 서로소 집합 자료구조의 정의와 구현

[Algorithm] 32강 : 최단 경로 알고리즘 기초 문제 풀이

[Algorithm] 31강 : 플로이드 워셜 알고리즘의 정의와 구현

[Algorithm] 30강 : 다익스트라 최단 경로 알고리즘의 정의와 구현

[Algorithm] 29강 : 다이나믹 프로그래밍의 기초 문제 풀이

python 서로소 python 서로소 집합 자료구조 서로소 집합 서로소 집합 자료구조

나아무늘보

혼자 끄적끄적하는 블로그 입니다.