SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

삼성 청년 SW 아카데미

APS 응용

SAMSUNG SW ACADEMY FOR YOUTH

목차

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

서로소 집합

서로소 집합(Disjoint-set)

- 서로소 또는 상호배타 집합들은 서로 중복 포함된 원소가 없는 집합들이다. 다시 말해 교집합이 없다.
- 집합에 속한 하나의 특정 멤버를 통해 각 집합들을 구분한다.
  이를 대표자(representative)라 한다.
- 서로소 집합을 표현하는 방법
  - 연결 리스트
  - 트리
- 서로소 집합 연산
  - Make-Set(x)
  - Find-Set(x)
  - Union(x , y)

서로소 집합(Disjoint-set)

서로소 집합 예

Make-Set(x)

Make-Set(y)

Make-Set(a)

Make-Set(b)

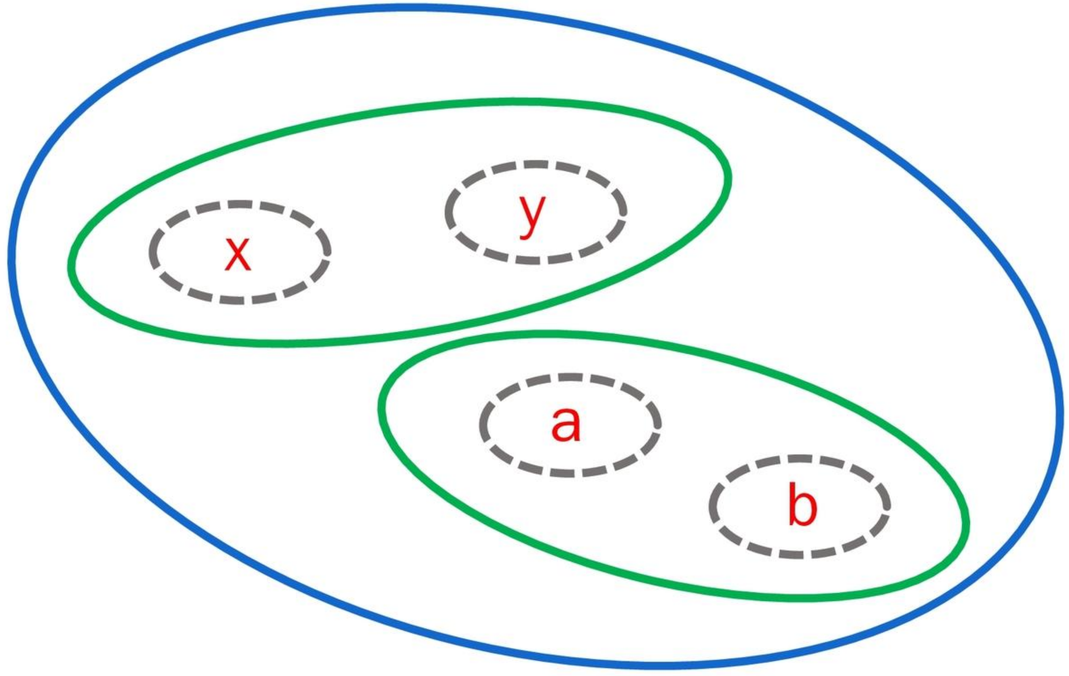
Union(x, y)

Union(a, b)

Find-Set(y) return x (representative)

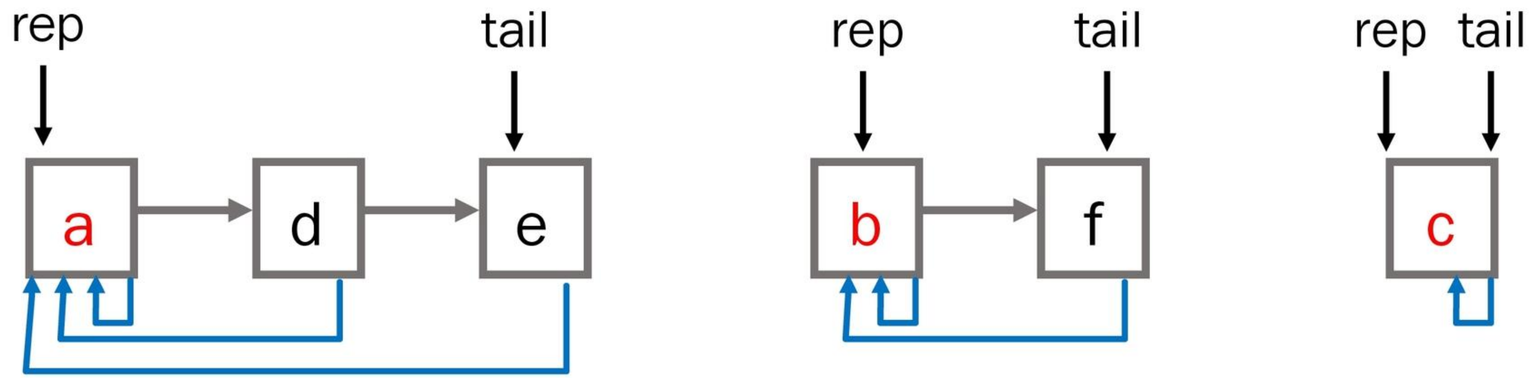
Find-Set(b) return a (representative)

Union(x, a)



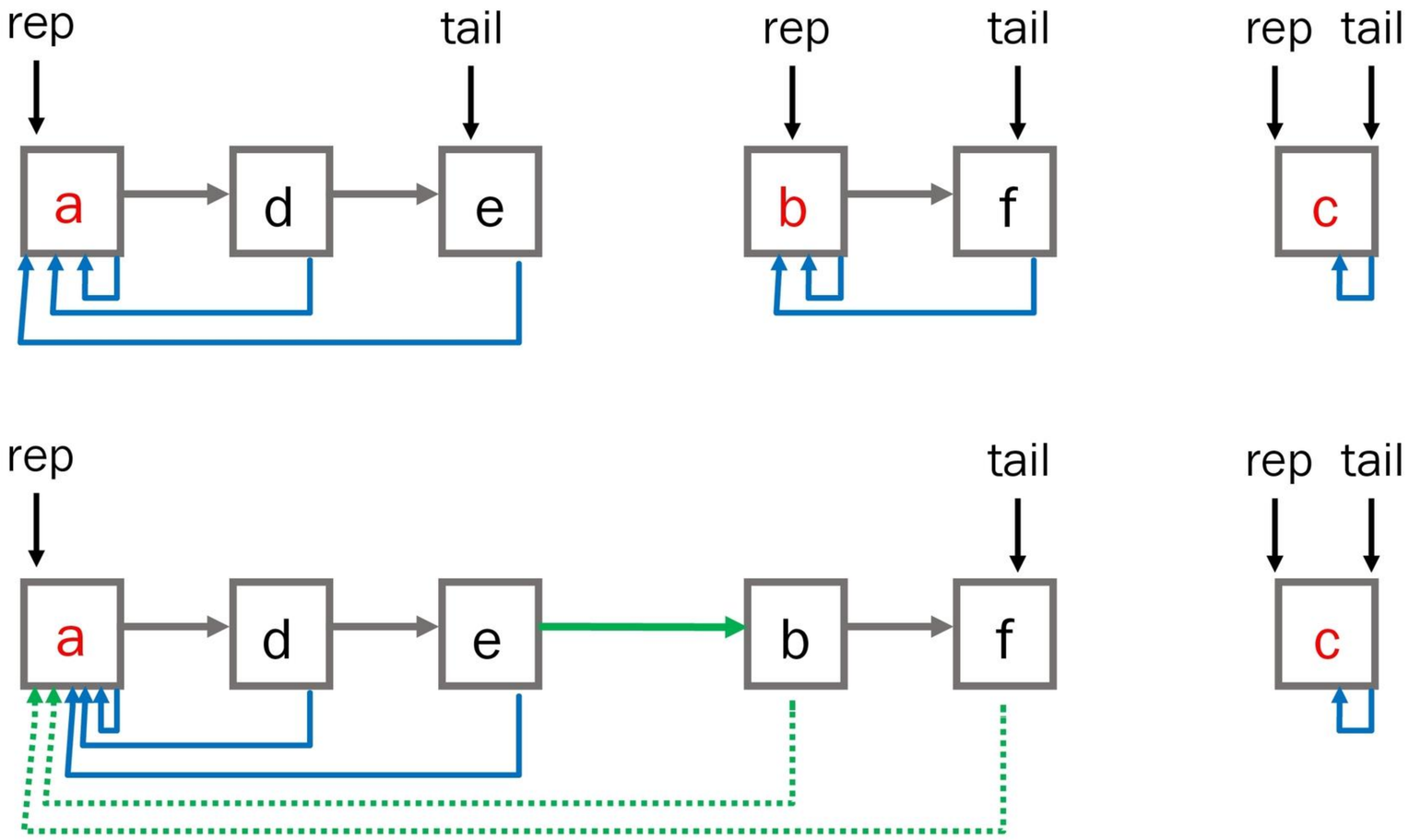
서로소 집합 표현 - 연결리스트

- 같은 집합의 원소들은 하나의 연결리스트로 관리한다.
- 연결리스트의 맨 앞의 원소를 집합의 대표 원소로 삼는다.
- 각 원소는 집합의 대표원소를 가리키는 링크를 갖는다.



서로소 집합 표현 - 연결리스트

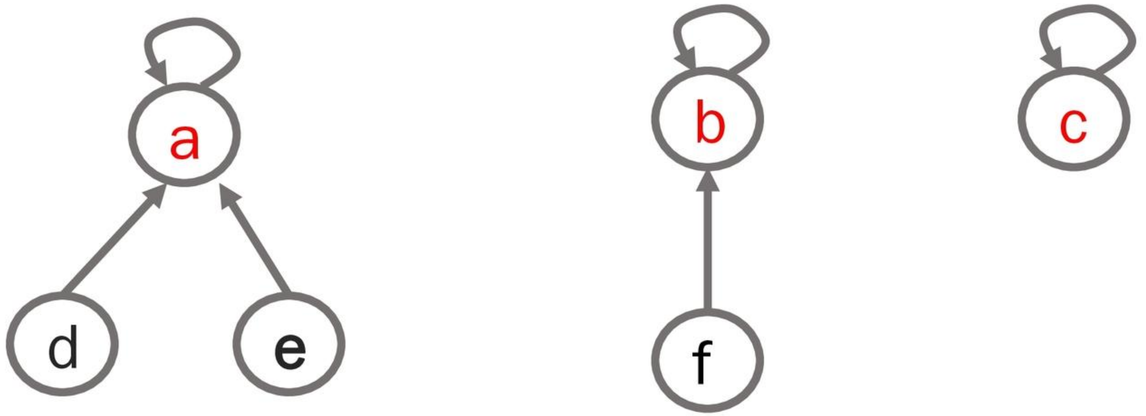
- 연결리스트 연산 예
  - Find-Set(e) return a
  - Find-Set(f) return b
  - Union(a, b)



서로소 집합 표현 - 트리

- 같은 집합의 원소들을 하나의 트리로 표현한다.

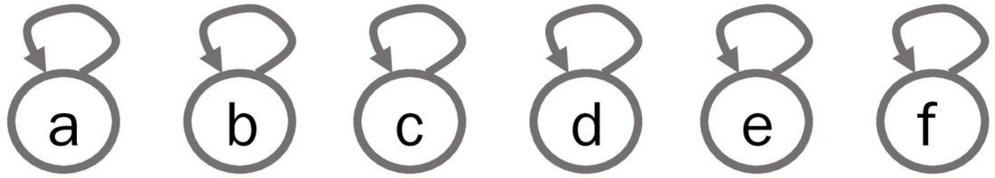
- 자식 노드가 부모 노드를 가리키며 루트 노드가 대표자가 된다.



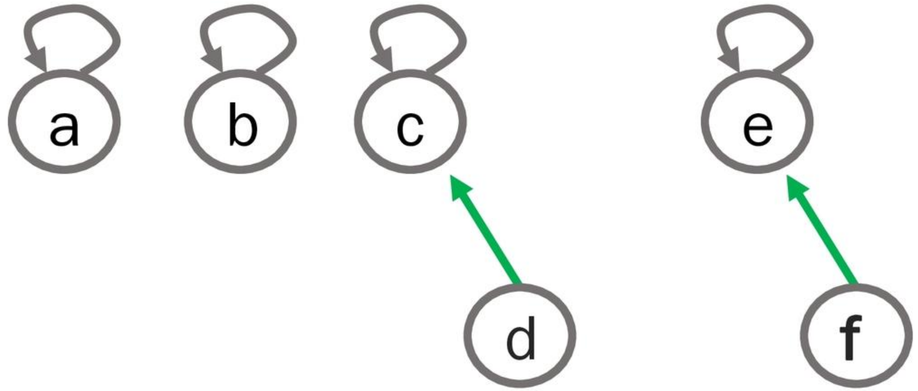
서로소 집합 표현 - 트리

연산 예

- Make-Set(a) ~ Make-Set(f)

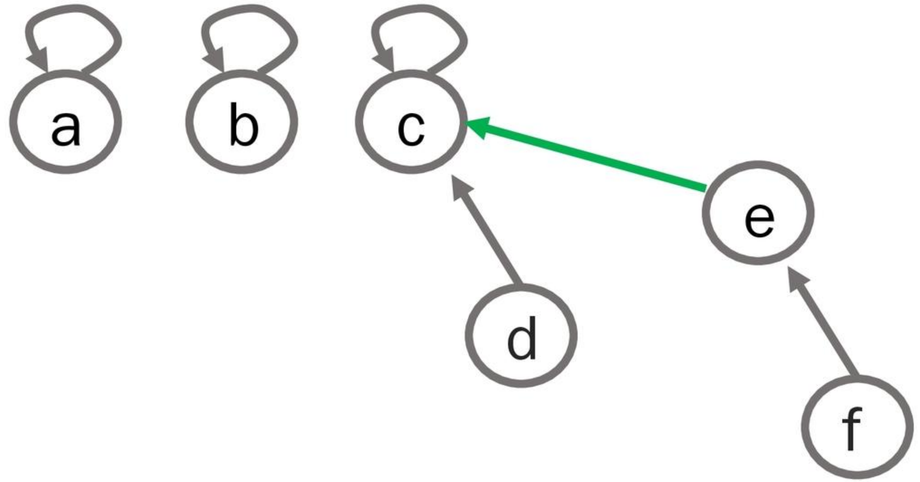


- Union(c, d), Union(e, f)



서로소 집합 표현 - 트리

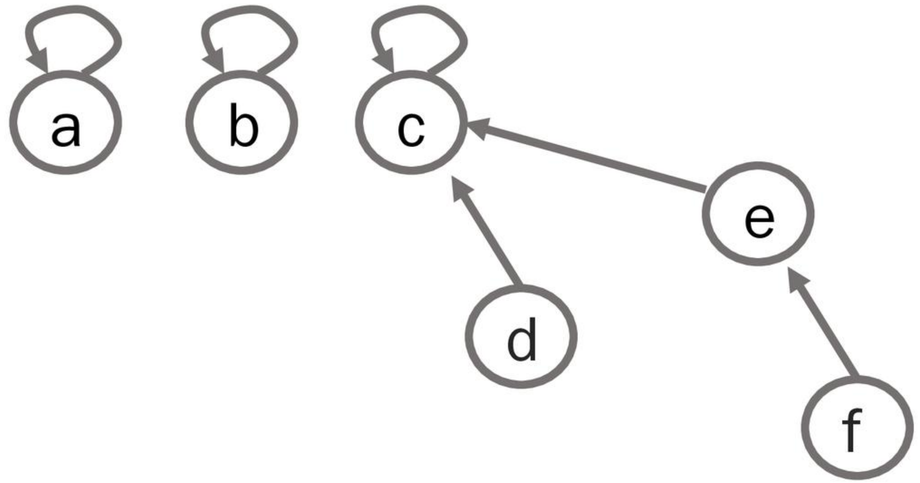
- Union(d, f)



- Find-Set(d) return c
- Find-Set(e) return c

서로소 집합 표현 - 트리

- 서로소 집합을 표현한 트리의 배열을 이용한 저장된 모습



인덱스	0	1	2	3	4	5
노드	a	b	c	d	e	f
부모 인덱스	0	1	2	2	2	4

서로소 집합에 대한 연산

- Make-Set(x) : 유일한 멤버 x를 포함하는 새로운 집합을 생성하는 연산

```
Make-Set(x)
        p[x] ← x
```

- Find_Set(x) : x를 포함하는 집합을 찾는 연산

```
Find-Set(x)
        IF x == p[x] : RETURN x
        ELSE         : RETURN Find_Set(p[x])
```

- Union(x , y) : x와 y를 포함하는 두 집합을 통합하는 연산

```
Union(x, y)
        IF Find-Set(y) == Find-Set(x) RETURN;
        p[Find-Set(y)] ← Find-Set(x)
```

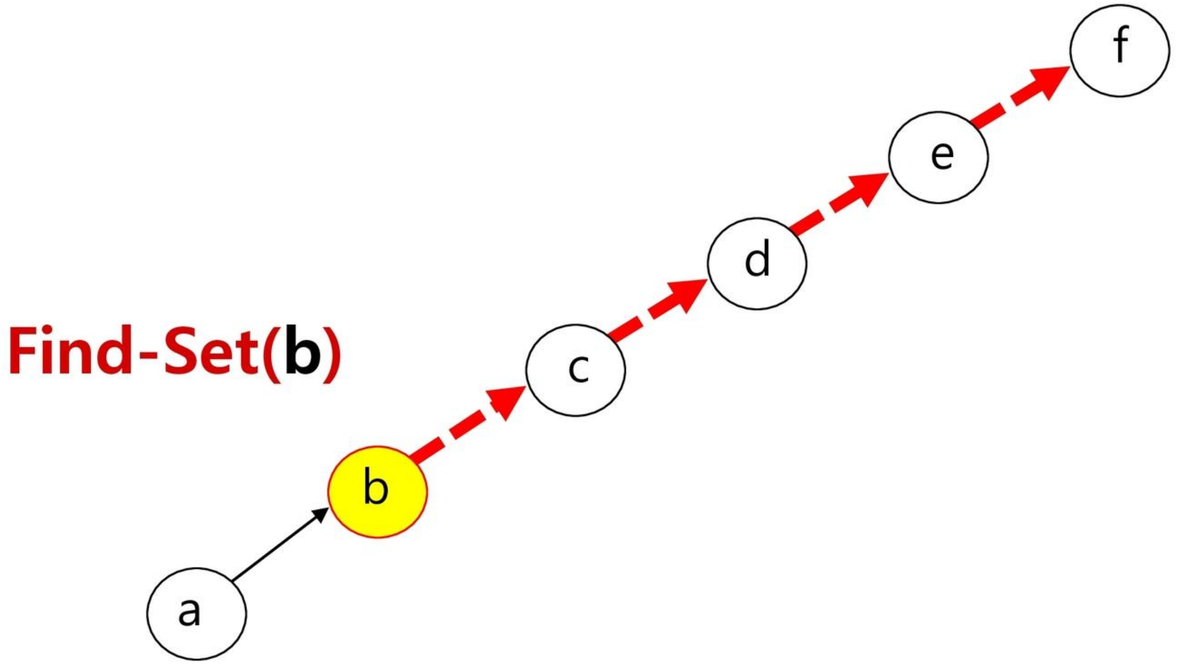
SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

서로소 집합 - 최적화

서로소 집합 - 최적화

- 문제점



서로소 집합 - 최적화

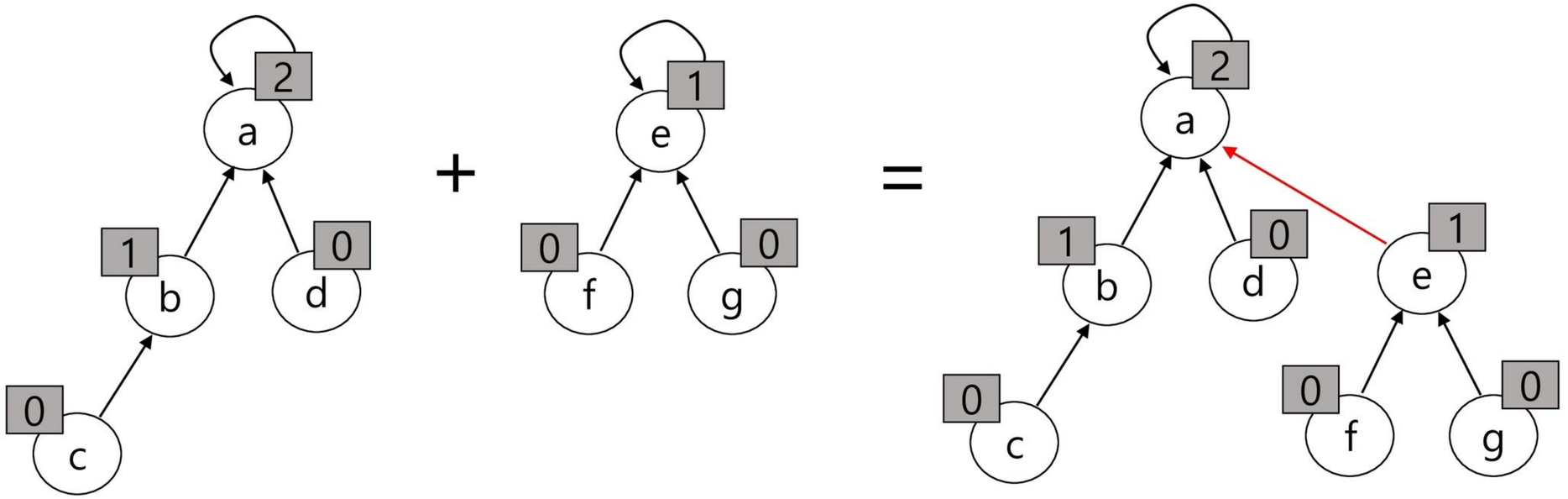
연산의 효율을 높이는 방법

- Rank를 이용한 Union
  - 각 노드는 자신을 루트로 하는 subtree의 높이를 rank로 저장한다.
  - 두 집합을 합칠 때 rank가 낮은 집합을 rank가 높은 집합에 붙인다.

- Path compression
  - Find-Set을 행하는 과정에서 만나는 모든 노드들이 직접 root를 가리키도록 포인터를 바꾸어 준다.

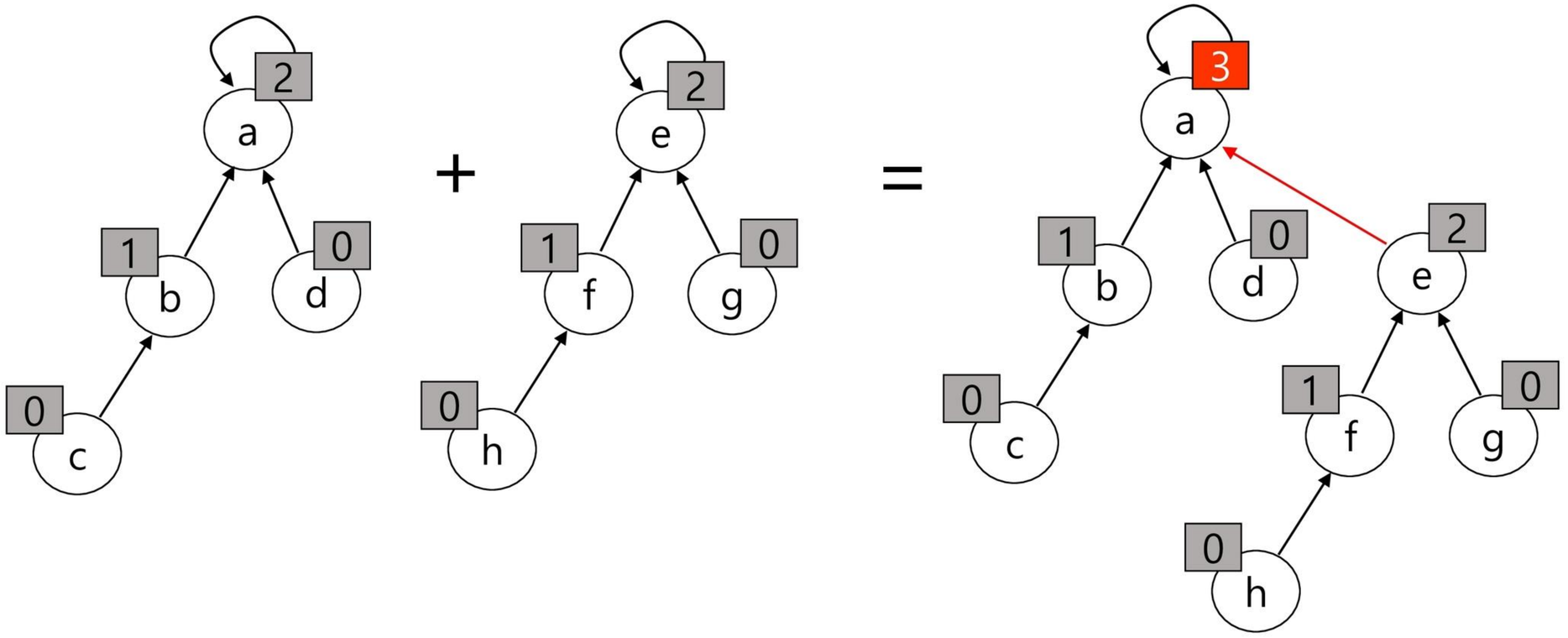
서로소 집합 - 최적화

- 랭크를 이용한 Union



서로소 집합 - 최적화

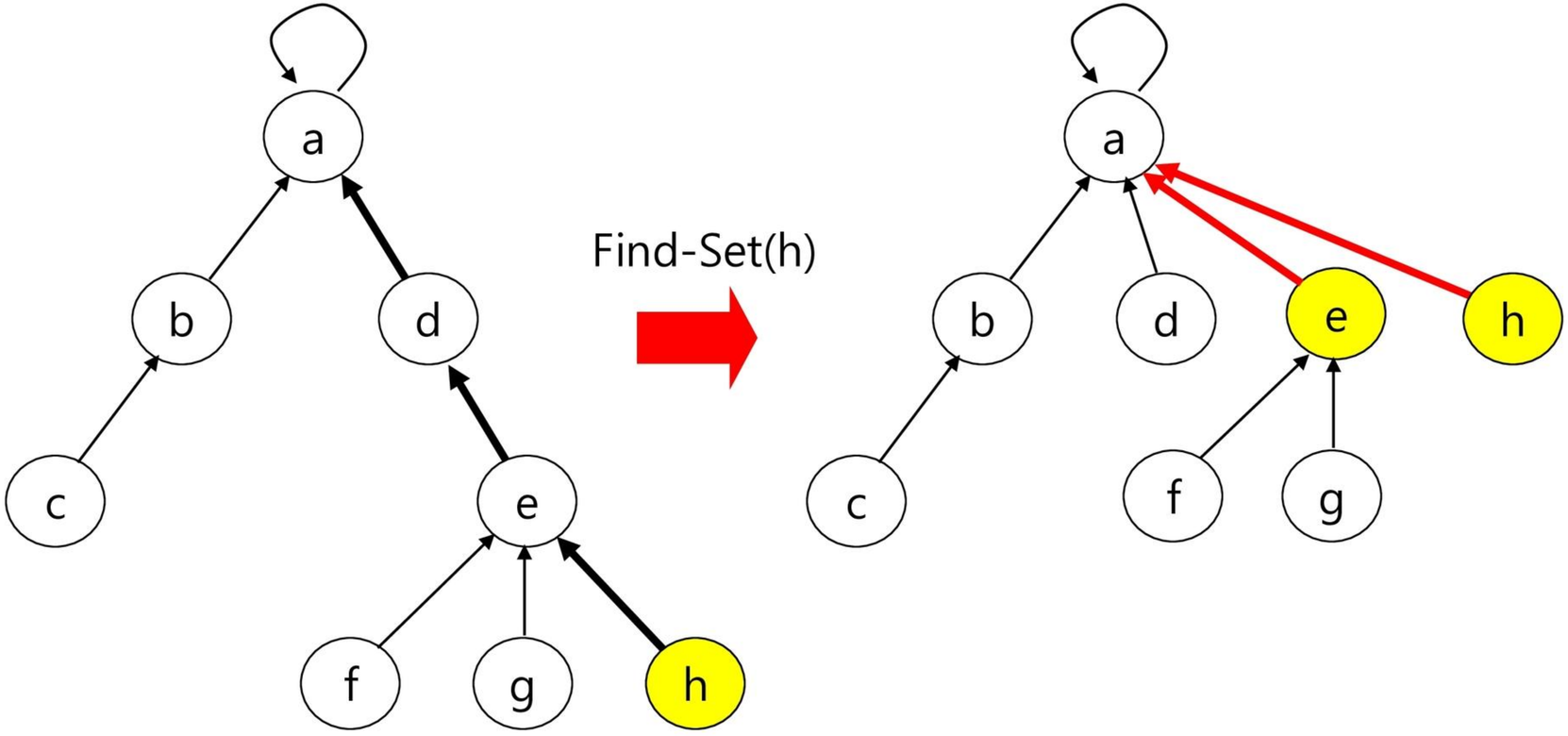
- 랭크를 이용한 Union에서 랭크가 증가하는 예



서로소 집합 - 최적화

Path Compression

- Path Compression을 적용한 Find-Set 연산은 특정 노드에서 루트까지의 경로를 찾아 가면서 노드의 부모 정보를 갱신한다.



서로소 집합 - 최적화

Path Compression 적용한 Find-Set 연산

- Find-Set(x) : x를 포함하는 집합을 찾는 오퍼레이션

전

```
Find-Set(x)
        IF x == p[x] : RETURN x
        ELSE         : RETURN Find-Set(p[x])
```

후

```
Find-Set(x)
        IF x == p[x] : RETURN x
        ELSE         : RETURN p[x] = Find-Set(p[x])
```

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

다음 방송에서 만나요!

삼성 청년 SW 아카데미