

## 친구인가? (Disjoint-Set : Union&Find)

오늘은 새 학기 새로운 반에서 처음 시작하는 날이다. 현수네 반 학생은 N명이다. 현수는 각 학생들의 친구관계를 알고 싶다.

모든 학생은 1부터 N까지 번호가 부여되어 있고, 현수에게는 각각 두 명의 학생은 친구 관계가 번호로 표현된 숫자쌍이 주어진다. 만약 (1, 2), (2, 3), (3, 4)의 숫자쌍이 주어지면 1번 학생과 2번 학생이 친구이고, 2번 학생과 3번 학생이 친구, 3번 학생과 4번 학생이 친구이다. 그리고 1번 학생과 4번 학생은 2번과 3번을 통해서 친구관계가 된다.

학생의 친구관계를 나타내는 숫자쌍이 주어지면 특정 두 명이 친구인지를 판별하는 프로그램을 작성하세요. 두 학생이 친구이면 "YES"이고, 아니면 "NO"를 출력한다.

### ■ 입력설명

매개변수 n에 반 학생수인 자연수  $N(1 \leq N \leq 1,000)$ 이 주어집니다.

매개변수 nums에  $M(1 \leq M \leq 3,000)$ 개의 숫자쌍이 주어집니다.

매개변수 s1, s2에 두 학생이 친구인지 확인해야 하는 학생번호가 주어집니다.

### ■ 출력설명

"YES"또는 "NO"를 반환합니다.

### ■ 매개변수 형식 1

9, [[1, 2], [2, 3], [3, 4], [1, 5], [6, 7], [7, 8], [8, 9]], 3, 8

### ■ 반환값 형식 1

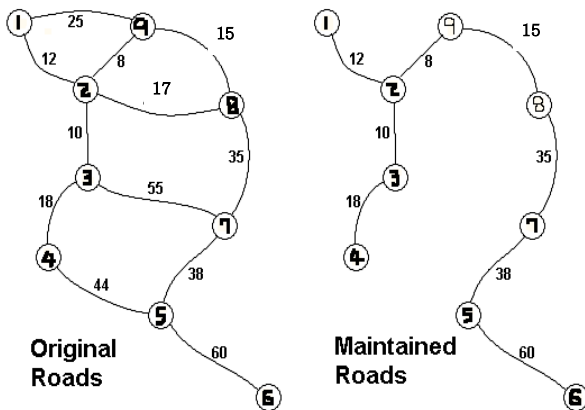
NO

## 원더랜드(최소스패닝트리 : 크루스칼, Union&amp;Find 활용)

원더랜드에 문제가 생겼다. 원더랜드의 각 도로를 유지보수하는 재정이 바닥난 것이다.

원더랜드는 모든 도시를 서로 연결하면서 최소의 유지비용이 들도록 도로를 선택하고 나머지 도로는 폐쇄하려고 한다.

아래의 그림은 그 한 예를 설명하는 그림이다.



위의 지도는 각 도시가 1부터 9로 표현되었고, 지도의 오른쪽은 최소비용 196으로 모든 도시를 연결하는 방법을 찾아낸 것이다.

#### 입력설명

매개변수  $n$ 에 도시의 개수  $N(1 \leq N \leq 100)$ 이 주어집니다.

매개변수  $edges$ 에  $E(1 \leq E \leq 1,000)$  개의 각 도로에 대한 정보를 나타내는 세 정수  $A, B, C$ 의 순서쌍 정보가 주어집니다. 이는  $A$ 번 도시와  $B$ 번 도시가 유지비용이  $C$ 인 도로로 연결되어 있다는 의미이다.

#### 출력설명

모든 도시를 연결하면서 드는 최소비용을 반환합니다.

#### 매개변수 형식 1

9, [[1, 2, 12], [1, 9, 25], [2, 3, 10], [2, 8, 17], [2, 9, 8], [3, 4, 18], [3, 7, 55], [4, 5, 44], [5, 6, 60], [5, 7, 38], [7, 8, 35], [8, 9, 15]]

#### 반환값 형식 1

196

## Trie 자료구조

```
class Node{
    constructor(){
        this.end = false;
        this.child = {}
    }
}

class Trie{
    constructor(){
        this.root=new Node();
    }
    insert(word){
        let cur=this.root;
        for(let x of word){
            if(cur.child[x]===undefined){
                cur.child[x]=new Node();
            }
            cur=cur.child[x];
        }
        cur.end=true;
    }
    search(word){
        let cur=this.root;
        for(let x of word){
            if(cur.child[x]===undefined) return false;
            cur=cur.child[x];
        }
        return cur.end;
    }
    prefixS(str){
        let cur=this.root;
        for(let x of str){
            if(cur.child[x]===undefined) return false;
            cur=cur.child[x];
        }
        return true;
    }
}
```

## 자동완성

포털 다음에서 검색어 자동완성 기능을 넣고 싶은 라이언은 한 번 입력된 문자열을 학습해서 다음 입력 때 활용하고 싶어 졌다. 예를 들어, go 가 한 번 입력되었다면, 다음 사용자는 g 만 입력해도 go를 추천해주므로 o를 입력할 필요가 없어진다! 단, 학습에 사용된 단어들 중 앞부분이 같은 경우에는 어쩔 수 없이 다른 문자가 나올 때까지 입력을 해야 한다.

효과가 얼마나 좋을지 알고 싶은 라이언은 학습된 단어들을 찾을 때 몇 글자를 입력해야 하는지 궁금해졌다.

예를 들어, 학습된 단어들이 아래와 같을 때

go

gone

guild

go를 찾을 때 go를 모두 입력해야 한다.

gone을 찾을 때 gon 까지 입력해야 한다. (gon이 입력되기 전까지는 go 인지 gone인지 확신할 수 없다.)

guild를 찾을 때는 gu 까지만 입력하면 guild가 완성된다.

이 경우 총 입력해야 할 문자의 수는 7이다.

라이언을 도와 위와 같이 문자열이 입력으로 주어지면 학습을 시킨 후, 학습된 단어들을 순서대로 찾을 때 몇 개의 문자를 입력하면 되는지 계산하는 프로그램을 만들어보자.

### ■ 입력설명

학습과 검색에 사용될 중복 없는 단어 N개가 주어진다.

모든 단어는 알파벳 소문자로 구성되며 단어의 수 N과 단어들의 길이의 총합 L의 범위는 다음과 같다.

$2 \leq N \leq 100,000$

$2 \leq L \leq 1,000,000$

### ■ 출력설명

단어를 찾을 때 입력해야 할 총 문자수를 리턴한다.

### ■ 매개변수 형식 1

["go","gone","guild"]

### ■ 반환값 형식 1

7

출처 : 프로그래머스

## 구간합 구하기(인덱스 트리)

어떤 N개의 수가 주어져 있다. 그런데 중간에 수의 변경이 빈번히 일어나고 그 중간에 어떤 부분의 합을 구하려 한다. 만약에 1,2,3,4,5 라는 수가 있고, 3번째 수를 6으로 바꾸고 2번째 부터 5번째까지 합을 구하라고 한다면 17을 출력하면 되는 것이다. 그리고 그 상태에서 다섯 번째 수를 2로 바꾸고 3번째부터 5번째까지 합을 구하라고 한다면 12가 될 것이다.

### ■ 입력설명

매개변수 nums에 N개의 수열이 주어집니다. 매개변수 order에 [a, b, c]로 이루어진 순서쌍이 주어집니다. a가 1이면 b번째 수를 c로 바꾸라는 의미입니다. a가 2이면 b번째부터 c번째까지의 수들의 합을 구하라는 의미입니다.

입력으로 주어지는 모든 수는  $-2^{63}$ 보다 크거나 같고,  $2^{63}-1$ 보다 작거나 같은 정수이다.

### ■ 출력설명

각각 차례대로 구한 구간합을 반환합니다. 단, 정답은  $-2^{63}$ 보다 크거나 같고,  $2^{63}-1$ 보다 작거나 같은 정수이다.

### ■ 매개변수 형식 1

[1, 2, 3, 4, 5], [[1, 3, 6], [2, 2, 5], [1, 5, 2], [2, 3, 5]]

### ■ 반환값 형식 1

[17, 12]