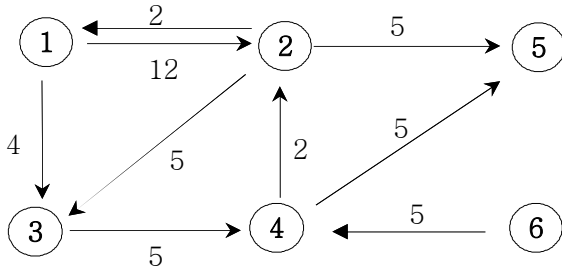


## 다익스트라 알고리즘

아래의 가중치 방향그래프에서 1번 정점(출발정점)에서 특정 정점(도착정점)까지의 최소 거리 비용을 출력하는 프로그램을 작성하세요. (경로가 없으면 -1을 출력한다)



### 입력설명

매개변수  $n$ 에 정점의 수  $N(1 \leq N \leq 20)$  주어지고, 매개변수 `edges`에 간선정보가 주어집니다.

[1, 2, 12]는 1번 정점에서 2번 정점으로 가는 비용이 12라는 뜻입니다. 각 간선의 비용값은 20을 넘지 않습니다.

매개변수 `end`에 도착정점이 주어집니다.

### 출력설명

1번 정점에서 도착정점까지의 최소비용을 반환합니다. 1번정점에서 도착정점까지 경로가 없다면 -1을 반환합니다.

### 매개변수 형식 1

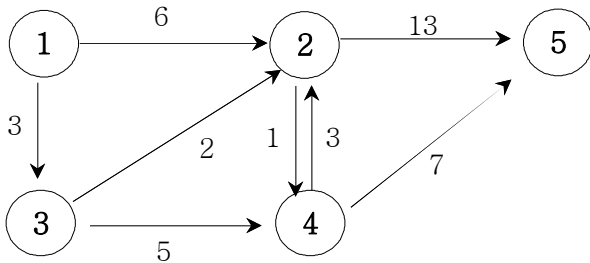
6, [[1, 2, 12], [1, 3, 4], [2, 1, 2], [2, 3, 5], [2, 5, 5], [3, 4, 5], [4, 2, 2], [4, 5, 5], [6, 4, 5]], 5

### 반환값 형식 1

14

## 플로이드 와샬 알고리즘

N개의 도시가 주어지고, 각 도시들을 연결하는 도로와 해당 도로를 통행하는 비용이 주어질 때 모든 도시에서 모든 도시로 이동하는데 쓰이는 비용의 최소값을 구하는 프로그램을 작성하세요.



### 입력설명

매개변수 n에 도시의 수  $N(N \leq 100)$ 과  $M(M \leq 200)$ 개의 간선 정보가 edges에 주어진다. 간선정보는 1번 도시와 2번도시가 연결되고 그 비용이 13이면 “1 2 13”으로 주어진다.

### 출력설명

모든 도시에서 모든 도시로 이동하는데 드는 최소 비용을 아래와 같이 출력해보세요. 자기자신으로 가는 비용은 0입니다. i번 정점에서 j번 정점으로 갈 수 없을 때는 비용을 “M”으로 합니다.

### 매개변수 형식 1

5, [[1, 2, 6], [1, 3, 3], [3, 2, 2], [2, 4, 1], [2, 5, 13], [3, 4, 5], [4, 2, 3], [4, 5, 7]]

### 출력 형식 1

0, 5, 3, 6, 13  
 M, 0, M, 1, 8  
 M, 2, 0, 3, 10  
 M, 3, M, 0, 7  
 M, M, M, M, 0

## 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 friend에  $M(1 \leq M \leq 3,000)$ 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

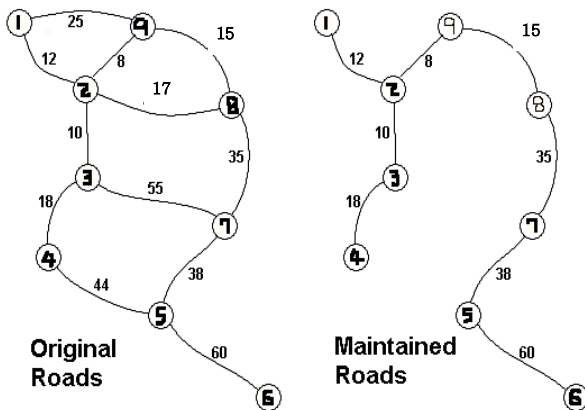
NO

## 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

## 동전교환(냅색 알고리즘)

다음과 같이 여러 단위의 동전들이 주어져 있을때 거스름돈을 가장 적은 수의 동전으로 교환 해주려면 어떻게 주면 되는가? 각 단위의 동전은 무한정 쓸 수 있다.

### ■ 입력설명

매개변수 coins에  $N(1 \leq N \leq 50)$ 개의 동전의 종류가 주어지고, 매개변수 m에 거슬러 줄 금액  $M(1 \leq M \leq 500)$ 이 주어진다.

각 동전의 종류는 100원을 넘지 않는다.

### ■ 출력설명

거슬러 줄 동전의 최소개수를 반환한다.

### ■ 매개변수 형식 1

[1, 2, 5], 15

### ■ 반환값 형식 1

3

출력설명 : 5 5 5 동전 3개로 거슬러 줄 수 있다.