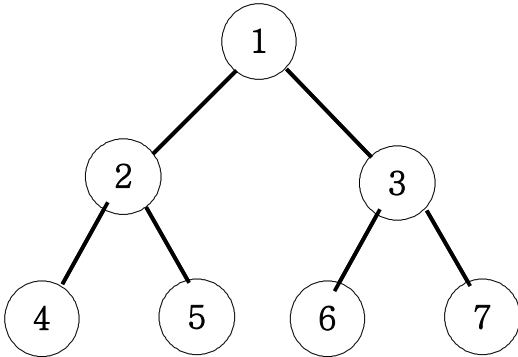


이진트리 레벨탐색(넓이우선탐색 : BFS)

아래 그림과 같은 이진트리를 큐(Queue) 자료구조를 이용해 레벨탐색을 해보세요.



레벨탐색 출력 : 1 2 3 4 5 6 7

아래 코드는 Queue 자료구조를 이용해 위에 트리를 레벨탐색한 코드입니다. 스스로 분석해 보세요.

```
function solution(){
  let answer="";
  function BFS(){
    let queue=[];
    queue.push(1);
    let L = 0;
    while(queue.length){
      let len = queue.length;
      answer += L + " : ";
      for(let i = 0; i < len; i++){
        let v = queue.shift();
        answer += v + " ";
        for(let nv of [v*2, v*2+1]){
          if(nv > 7) continue;
          queue.push(nv);
        }
      }
      console.log(answer);
      answer = "";
      L++;
    }
  }
  BFS();
  return;
}
solution();
```

송아지 찾기(BFS : 상태트리탐색)

현수는 송아지를 잃어버렸다. 다행히 송아지에는 위치추적기가 달려 있다. 현수의 위치와 송아지의 위치가 수직선상의 좌표 점으로 주어지면 현수는 현재 위치에서 송아지의 위치까지 다음과 같은 방법으로 이동한다. 송아지는 움직이지 않고 제자리에 있다.

현수는 스카이 쿱을 타고 가는데 한 번의 점프로 앞으로 1, 뒤로 1, 앞으로 5를 이동할 수 있다. 최소 몇 번의 점프로 현수가 송아지의 위치까지 갈 수 있는지 구하는 프로그램을 작성하세요.

■ 입력설명

매개변수 s에 현수의 위치 S와 매개변수 e에 송아지의 위치 E가 주어집니다.
직선의 좌표 점은 1부터 10,000까지이다.

■ 출력설명

점프의 최소횟수를 반환합니다. 답은 1이상입니다.

■ 매개변수 형식 1

5 14

■ 반환값 형식 1

3

■ 매개변수 형식 2

8 3

■ 반환값 형식 2

5

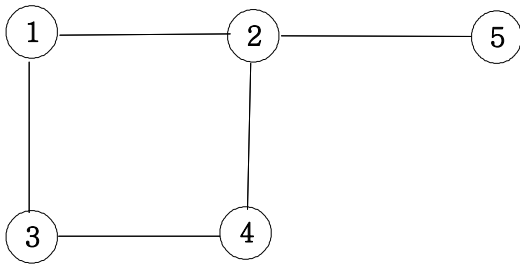
그래프

- 그래프는 $G(V, E)$ 로 정의하고, V (Vertex : 정점)과 E (Edge : 간선)의 집합을 의미한다.
- 연결되어 있는 원소들간의 관계를 표현하는 자료구조이다.

1. 인접행렬 : 배열을 이용해 그래프를 표현하는 방법

1) 무방향 그래프

graph



	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	1
3	1	0	0	1	0
4	0	1	1	0	0
5	0	1	0	0	0

입력형식 :

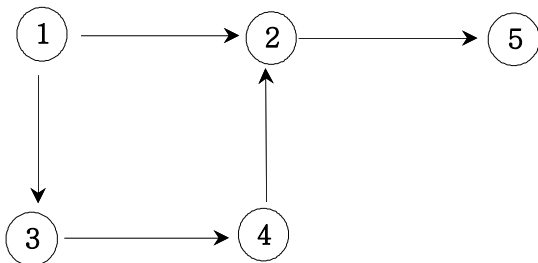
edge = [[1, 2], [1, 3], [2, 4], [2, 5], [3, 4]]

```

let graph=Array.from(Array(n+1), ()=>Array(n+1).fill(0));
for(let [a, b] of edges){
    graph[a][b]=1;
    graph[b][a]=1;
}
  
```

2) 방향그래프

graph

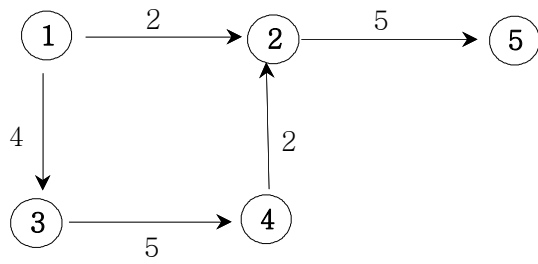


	1	2	3	4	5
1	0	1	1	0	0
2	0	0	0	0	1
3	0	0	0	1	0
4	0	1	0	0	0
5	0	0	0	0	0

입력형식 :

edge = [[1, 2], [1, 3], [2, 5], [3, 4], [4, 2]]

3) 가중치 방향그래프



graph

	1	2	3	4	5
1	0	2	4	0	0
2	0	0	0	0	5
3	0	0	0	5	0
4	0	2	0	0	0
5	0	0	0	0	0

입력형식 :

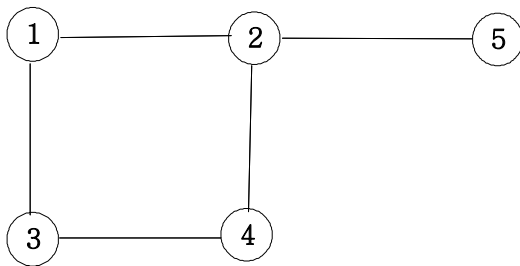
```
edge = [[1, 2, 2], [1, 3, 4], [2, 5, 5], [3, 4, 5], [4, 2, 2]]
```

```

let graph=Array.from(Array(n+ 1), ()=>Array(n+ 1).fill(0));
for(let [a, b, c] of edges){
    graph[a][b]=c;
}
  
```

1. 인접리스트 : 연결리스트를 이용해 그래프를 표현하는 방법

1) 무방향 그래프



graph

```

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

```

입력형식 :

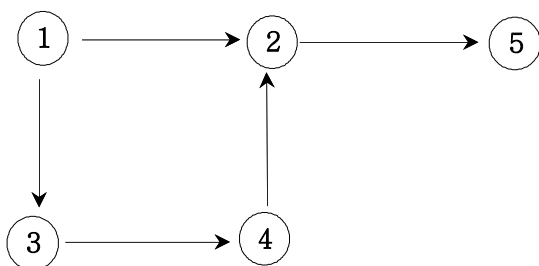
```
edge = [[1, 2], [1, 3], [2, 4], [2, 5], [3, 4]]
```

```

let graph=Array.from(Array(n+ 1), ()=>Array());
for(let [a, b] of edges){
    graph[a].push(b);
    graph[b].push(a);
}

```

2) 방향그래프



graph

```

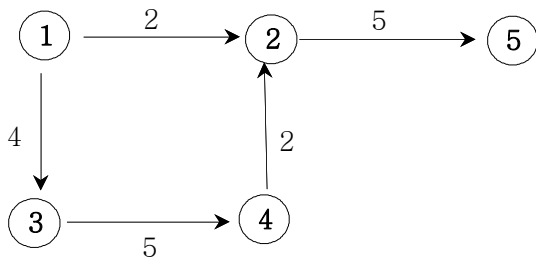
[
  []
  [2, 3]
  [5]
  [4]
  [2]
  []
]

```

입력형식 :

```
edge = [[1, 2], [1, 3], [2, 5], [3, 4], [4, 2]]
```

3) 가중치 방향그래프



graph

```

[
  []
  [(2, 2), (3, 4)]
  [(5, 5)]
  [(4, 5)]
  [(2, 2)]
  []
]

```

입력형식 :

```
edge = [[1, 2, 2], [1, 3, 4], [2, 5, 5], [3, 4, 5], [4, 2, 2]]
```

```

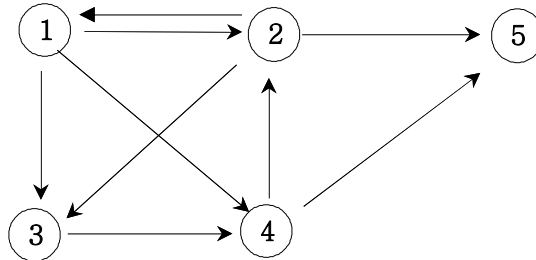
let graph=Array.from(Array(n+ 1), ()=>Array());
for(let [a, b, c] of edges){
    graph[a].push([b, c]);
}

```

경로 탐색(인접행렬)

방향그래프가 주어지면 1번 정점에서 N번 정점으로 가는 모든 경로의 가지 수를 출력하는 프로그램을 작성하세요. 아래 그래프에서 1번 정점에서 5번 정점으로 가는 가지 수는

1 2 3 4 5
 1 2 5
 1 3 4 2 5
 1 3 4 5
 1 4 2 5
 1 4 5



총 6 가지입니다.

입력설명

매개변수 $n(1 \leq n \leq 20)$ 과 매개변수 `edge`에 간선의 연결정보가 주어진다.

출력설명

총 경로의 가지수를 반환한다.

입력예제 1

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

출력예제 1

6