

특정 거리의 도시 찾기

백준 18352

202021018 오윤주

문제

특정 거리의 도시 찾기 성공



2 실버 II

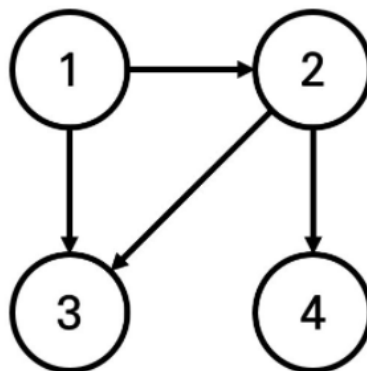
시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	256 MB	17422	5162	3273	28.416%

문제

어떤 나라에는 1번부터 N 번까지의 도시와 M 개의 단방향 도로가 존재한다. 모든 도로의 거리는 1이다.

이 때 특정한 도시 X 로부터 출발하여 도달할 수 있는 모든 도시 중에서, 최단 거리가 정확히 K 인 모든 도시들의 번호를 출력하는 프로그램을 작성하시오. 또한 출발 도시 X 에서 출발 도시 X 로 가는 최단 거리는 항상 0이라고 가정한다.

예를 들어 $N=4$, $K=2$, $X=1$ 일 때 다음과 같이 그래프가 구성되어 있다고 가정하자.



이 때 1번 도시에서 출발하여 도달할 수 있는 도시 중에서, 최단 거리가 2인 도시는 4번 도시 뿐이다. 2번과 3번 도시의 경우, 최단 거리가 1이기 때문에 출력하지 않는다.

문제

첫째 줄에 도시의 개수 N , 도로의 개수 M , 거리 정보 K , 출발 도시의 번호 X 가 주어진다. ($2 \leq N \leq 300,000$, $1 \leq M \leq 1,000,000$, $1 \leq K \leq 300,000$, $1 \leq X \leq N$) 둘째 줄부터 M 개의 줄에 걸쳐서 두 개의 자연수 A, B 가 공백을 기준으로 구분되어 주어진다. 이는 A 번 도시에서 B 번 도시로 이동하는 단방향 도로가 존재한다는 의미다. ($1 \leq A, B \leq N$) 단, A 와 B 는 서로 다른 자연수이다.

출력

X 로부터 출발하여 도달할 수 있는 도시 중에서, 최단 거리가 K 인 모든 도시의 번호를 한 줄에 하나씩 오름차순으로 출력한다.

이 때 도달할 수 있는 도시 중에서, 최단 거리가 K 인 도시가 하나도 존재하지 않으면 -1을 출력한다.

예제 입력 1 복사

```
4 4 2 1
1 2
1 3
2 3
2 4
```

예제 출력 1 복사

```
4
```

구현

```
from collections import deque

n, m, k, x = map(int, input().split())
graph = [[] for _ in range(n+1)]
check = [-1] * (n+1) // 변수 입력받기, 간선과 확인한 도시
                        변수 만들기

def bfs(node):
    q = deque()
    q.append(node)
    check[node] = 0 // 0으로 바꾸기
    while q:
        node = q.popleft()
        for n in graph[node]:
            if check[n] == -1:
                check[n] = check[node] + 1 // 그래프 돌면서 확인하기
                q.append(n)
```

구현

```
for _ in range(m):  
    a, b = map(int, input().split())  
    graph[a].append(b)
```

// a, b 입력받아 그래프에 넣기

```
bfs(x)
```

```
if k not in check:  
    print(-1)
```

```
else:
```

```
    for i in range(n+1):  
        if check[i] == k:  
            print(i)
```

// 결과 출력

감사합니다.