

백준 18352
특정거리의 도시 찾기

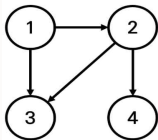
202020988 조아영

001 문제 소개

002 아이디어

003 코드 설명

문제 소개



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.

입력 : 도시의 개수 N , 도로의 개수 M , 거리정보 K , 출발도시의 번호 X
($2 \leq N \leq 300,000$, $1 \leq M \leq 1,000,000$, $1 \leq K \leq 300,000$, $1 \leq X \leq N$)
 M 개의 줄에 걸친 A, B ($A \rightarrow B$ 도로 존재, $1 \leq A, B, \leq N$)

출력 : X 에서 출발하여 도달할 수 있는 도시 중 최단거리가 K 인 도시의 번호 오름차순으로 출력, 초 1 | 단거리가 K 인 도시가 없는 경우 -1 출력

아이디어

각 도시의 연결 정보를 2차원 배열(인접리스트)에 저장



각 노드를 BFS로 탐색하며
1차원 배열에 노드마다 최단거리 저장해놓음

코드 설명

```
import sys
from collections import deque

N, M, K, X = map(int, sys.stdin.readline().split())  입력

q = deque()  BFS 사용 위한 queue
graph = [[] for i in range(N+1)]  연결된 도시 정보 저장하기 위해 초기화
visited = [0 for i in range(N+1)]  도시 방문 여부와 최단거리 저장 위해 초기화

for _ in range(M):
    A, B = map(int, sys.stdin.readline().split())
    graph[A].append(B)  연결된 도시 정보 저장

q.append(X)  첫번째 방문 도시 삽입
```

코드 설명

```
while q: queue가 비었을 때까지 반복
    temp = q.popleft()
    for i in graph[temp]: temp노드에 연결된 도시들 반복
        if visited[i] == 0 and i != X: 방문한 적 없고, 첫번째 방문한 도시가 아닌경우
            visited[i] = visited[temp] + 1
            q.append(i)                최단거리 정보 저장
            queue에 노드 삽입

flag = 0
for i in range(N+1):
    if visited[i] == K: 최단거리가 K인 노드 출력
        print(i)
        flag = 1

if flag == 0: 최단거리가 K인 노드가 없는 경우 -1 출력
    print("-1")
```

감사합니다