Algorithm

[Algorithm] 32강 : 최단 경로 알고리즘 기초 문제 풀이

2020. 11. 20. 16:24 수정 삭제 공개

<문제> 전보

- 어떤 나라에는 **N개의 도시**가 있다. 그리고 각 도시는 보내고자 하는 메시지가 있는 경우, 다른 도시로 전보를 보내서 다른 도시로 해당 메시지를 전송할 수 있다.
- 하지만 X라는 도시에서 Y라는 도시로 전보를 보내고자 한다면, 도시 X에서 Y로 향하는 통로가 설치되어 있어야 한다. 예를 들어 X에서 Y로 향하는 통로는 있지만, Y에서 X로 향하는 통로가 없다면 Y는 X로 메시지를 보낼 수 없다. 또한 통로를 거쳐 메시지를 보낼 때는 일정 시간이 소요된다.
- 어느 날 C라는 도시에서 위급 상황이 발생했다. 그래서 최대한 많은 도시로 메시지를 보내고자 한다. 메시지는 **도시 C에서 출발하여 각 도시 사이에 설치된 통로를 거쳐, 최대한 많이 퍼져나갈 것**이다.
- 각 도시의 번호와 통로가 설치되어 있는 정보가 주어졌을 때, 도시 C에서 보낸 메시지를 받게 되는 도시의 개수는 총 몇 개이며 도시들이 모두 메시지를 받는 데까지 걸리는 시간은 얼마인지 계산하는 프로그램을 작성하시오.

문제 해결 아이디어

- 핵심 아이디어 : 한 도시에서 다른 도시까지의 최단 거리 문제로 치환할 수 있다.

- N과 M의 범위가 충분히 크기 때문에 우선 순위 큐를 활용한 다익스트라 알고리즘을 구현

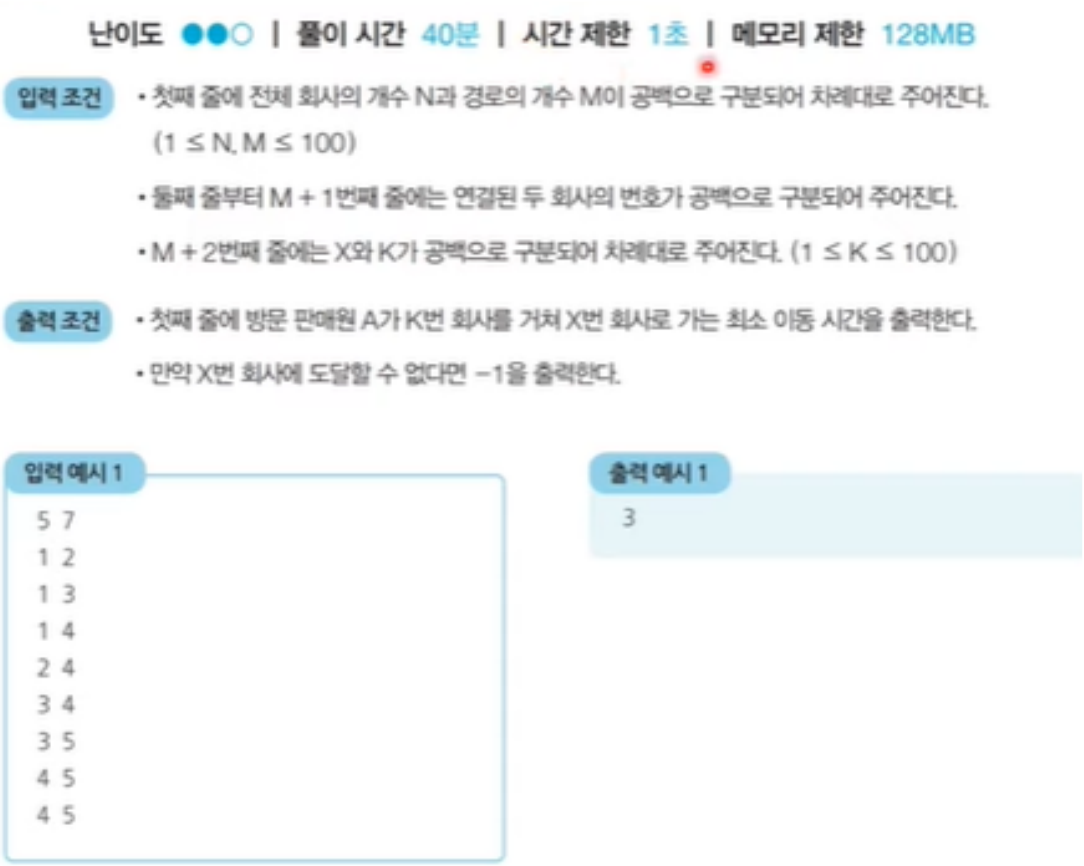
풀이 구현

```
import heapq
import sys
input = sys.stdin.readline
INF = int(1e9) # 무한을 의미하는 값으로 10억을 설정

def dijkstra(start):
    q = []
    # 시작 노드로 가기 위한 최단 거리는 0으로 설정하여, 큐에 삽입
    heapq.heappush(q,(0,start))
    distance[start] = 0
    while q: # 큐가 비어있지 않다면
        #가장 최단 거리가 짧은 노드에 대한 정보를 꺼내기
        dist, now = heapq.heappop(q)
        if distance[now] < dist:
            continue
        # 현재 노드와 연결된 다른 인접한 노드들을 확인
        for i in graph[now]:
            cost = dist +i[1]
            # 현재 노드를 거쳐서, 다른 노드로 이동하는 거리가 더 짧은 경우
            if cost < distance[i[0]]:
                distance[i[0]] = cost
                heapq.heappush(q,(cost,i[0]))

# 노드의 개수, 간선의 개수, 시작노드를 입력받기
n,m,start = map(int,input().split())

#각 노드에 연결되어 있는 노드에 대한 정보를 담는 리스트를 만들기
graph = [[] for i in range(n+1) ]

#최단 거리 테이블을 모두 무한으로 초기화
distance = [INF] * (n +1)

# 모든 간선 정보를 입력받기
for _ in range(m):
    x,y,z = map(int,input().split())
    # x 번 노드에서 y 번 노드로 가는 비용이 z 라는 의미
    graph[x].append((y,z))

# 다익스트라 알고리즘을 수행
dijkstra(start)

# 도달 할 수 있는 노드의 개수
count = 0
```

```
#도달 할 수 있는 노드중에서, 가장 멀리 있는 노드와의 최단거리
max_distance = 0
for d in distance = 0:
   # 도달할 수 있는 노드인 경우
   if d != 1e9:
      count += 1
      max_distance = max(max_distance,d)

# 시작 노드는 제외해야 하므로 count -1 을 출력
print(count-1, max_distance)
```

<문제> 미래도시

- 미래 도시에는 1번부터 N번까지의 회사가 있는데 특정 회사끼리는 서로 도로를 통해 연결되어 있다. 방문 판매원 A는 현재 1번 회사에 위치해 있으며, X번 회사에 방문해 물건을 판매하고자 한다.
- 미래 도시에서 특정 회사에 도착하기 위한 방법은 회사끼리 연결되어 있는 도로를 이용하는 방법이 유일하다. 또한 연결된 2개의 회사는 **양방향**으로 이동할 수 있다. 공중 미래 도시에서 특정 회사와 다른 회사가 도로로 연결되어 있다면, 정확히 1만큼의 시간으로 이동할 수 있다.
- 또한 오늘 방문 판매원 A는 기대하던 소개팅에도 참석하고자 한다. 소개팅의 상대는 K번 회사에 존재한다. 방문 판매원 A는 X번 회사에 가서 물건을 판매하기 전에 먼저 소개팅 상대의 회사에 찾아가서 함께 커피를 마실 예정이다. 따라서 방문 판매원 A는 **1번 회사에서 출발하여 K번 회사를 방문한 뒤에 X번 회사로 가는 것이 목표**다. 이때 방문 판매원 A는 가능한 한 빠르게 이동하고자 한다.
- 방문 판매원이 회사 사이를 이동하게 되는 **최소 시간**을 계산하는 프로그램을 작성하시오.



난이도 ●●○ | 풀이 시간 40분 | 시간 제한 1초 | 메모리 제한 128MB

입력 조건
- 첫째 줄에 전체 회사의 개수 N과 경로의 개수 M이 공백으로 구분되어 차례대로 주어진다. (1 ≤ N, M ≤ 100)
- 둘째 줄부터 M + 1번째 줄에는 연결된 두 회사의 번호가 공백으로 구분되어 주어진다.
- M + 2번째 줄에는 X와 K가 공백으로 구분되어 차례대로 주어진다. (1 ≤ K ≤ 100)

출력 조건
- 첫째 줄에 방문 판매원 A가 K번 회사를 거쳐 X번 회사로 가는 최소 이동 시간을 출력한다.
- 만약 X번 회사에 도달할 수 없다면 -1을 출력한다.

입력 예시 1

```
5 7
1 2
1 3
1 4
2 4
3 4
3 5
4 5
4 5
```

출력 예시 1

```
3
```

문제 해결 아이디어

- 핵심 아이디어: 전형적인 최단 거리 문제이므로 최단 거리 알고리즘을 이용해 해결
- N의 크기가 최대 100 이므로 플로이드 워셜 알고리즘을 이용 가능
- 플로이드 워셜 알고리즘을 수행한 뒤에 (1번 노드에서 X까지의 최단거리 + X에서 K까지의 최단 거리)를 계산하여 출력하면 정답

문제 구현

```
INF = int(1e9) # 무한을 의미하는 값으로 10억을 설정

# 노드의 개수 및 간선의 개수를 입력받기
n,m = map(int, input().split())
```

```python
# 2차원 리스트(그래프 표현)를 만들고, 모든 값을 무한으로 초기화
graph = [[INF] * (n+1) for _ in range(n+1)]

# 자기 자신에서 자기 자신으로 가는 비용은 0으로 초기화
for a in range(1,n+1):
    for b in range(1,n+1):
        if a==b:
            graph[a][b] = 0

# 각 간선에 대한 정보를 입력 받아, 그 값으로 초기화
for _ in range(m):
    # A와 B가 서로에게 가는 비용은 1이라고 설정
    a,b = map(int,input().split())
    graph[a][b] = 1
    graph[b][a] = 1

# 거쳐 갈 노드 X와 최종 목적지 노드 K를 입력받기
x, k = map(int,input().split())

# 점화식에 따라 플로이드 워셜 알고리즘을 수행
for k in range(1,n+1):
    for a in range(1,n+1):
        for b in range(1,n+1):
            graph[a][b] = min(graph[a][b], graph[a][k] + graph[k][b])

# 수행된 결과를 출력
distance = graph[1][k] + graph[k][x]

# 도달할 수 없는 경우, -1을 출력
if distance >= INF:
    print("-1")
# 도달할 수 있다면, 최단 거리를 출력
else:
    print(distance)
```


'Algorithm' 카테고리의 다른 글

[Algorithm] 32강 : 최단 경로 알고리즘 기초 문제 풀이

[Algorithm] 31강 : 플로이드 워셜 알고리즘의 정의와 구현

[Algorithm] 30강 : 다익스트라 최단 경로 알고리즘의 정의와 구현

[Algorithm] 29강 : 다이나믹 프로그래밍의 기초 문제 풀이

[Algorithm] 28강 : 다이나믹 프로그래밍의 정의와 구현

[Algorithm] 27강 : 이진 탐색 기초 문제 풀이

최단 경로 알고리즘 최단 경로 알고리즘 문제풀이 플로이드 워셜 알고리즘 풀이

나아무늘보

혼자 끄적끄적하는 블로그 입니다.