

다익스트라 알고리즘

- 다이나믹 프로그래밍 은 활용한 대표적인 최단 경로 탐색 알고리즘입니다.

하나의 정점에서 다른 모든 정점으로 가는 최단 경로

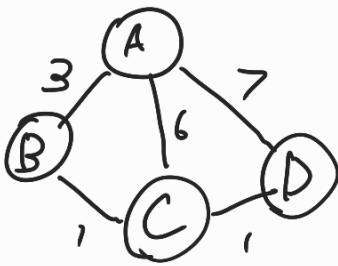
* 이 때 음의 간선은 포함 할 수 없다.

→ 0보다 작은 값을 가지는 간선 (노드 간의 거리)

- 현재까지 알고있던 최단 경로를 계속해서 갱신

작동 과정

1. 출발노드 설정
2. 출발노드 기준으로 각 노드의 최소 비용 저장
3. 방문하지 않은 노드 중 가장 비용이 적은 노드 선택
4. 해당 노드를 거쳐서 특정한 노드로 가는 경우 고려하여 최소비용 갱신



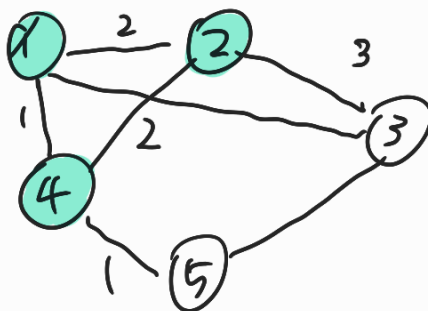
1. 출발노드 A로 설정

2. B: 3, C: 6 D: 7 거리 저장

3. B 선택

4. $B \rightarrow C = 3 + 1 = 4$ $A \rightarrow C > A \rightarrow B \rightarrow C$ 최소비용 갱신

* 3번의 경우 현재있는 노드의 기준이 아닌 Target node 기준이다



이 경우 지나간 노드이다.

현재 2번노드로 선택돼 있을 때 2번 기준 근접노드는 3번이지만 target node 기준 으로 다음노드는 5번이다.

Baek Joon 1753. 최단 경로

문제

방향 그래프가 주어지면 주어진 시작점에서 다른 모든 정점으로의 최단 경로를 구하는 프로그램을 작성하시오 (단 모든 간선의 가중치는 10 이하 자연수)

→ 간선에 방향성을 부여한 것 $A \rightarrow B$ 일때 A에서 B로는 갈 수 있지만 $B \rightarrow A$ 로는 불가능하다.

Input

정점의 개수, 간선의 개수
시작 정점의 번호

※ 서로 다른 두 정점 사이 여러개의 간선이 존재할 수도 있다.

간선의 개수 (from, to, 가중치(거리))

output

정점의 개수 (시작 정점에서 i번 정점까지의 최단 거리)
없을 경우 INF, 나 자신은 0

Program

1. Input 받기
2. 2차원 배열에 가중치 저장 (정렬 후 최소값만 저장)
3. 최소거리 저장 배열 (size 정점의 개수) 생성
4. dequeue 생성
5. while loop
 - 최소거리 update
 - 다음 노드로 이동

1753 신축상환

프로그램 상 구현은 성공 그러나 시간초과로 오답 처리

원인은 다음노드로 최소 distance node를 구할 때

$node = distance.index(\min(distance)) \leftarrow$ 여기로 이공간이 $O(N)$

결론: priority queue를 사용하여 구현

우선순위 큐 (priority queue)

- INSERT(x) : 새로운 원소 x 삽입
- Extract-MIN(): 최솟값을 삭제하고 반환. ※ Extract-MAX()도 존재

python library queue.PriorityQueue 사용시 시간초과 발생

↳ heapq 사용

