

Graph



백준 1197, 최소 스패닝 트리

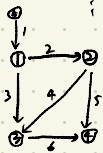
개념
크루스칼 알고리즘
Union-Find
반장게스론

입력
3(점 A) 3(점 B)
1(점 A) 2(점 B) 1(가중치)
2 3 2
1 3 3

백준 1753, 최단 거리

개념
다익스트라 알고리즘
최소 비용

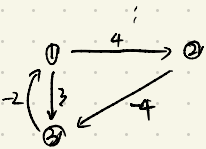
입력
5(점 A) 6(점 B)
1(가중치)
5(점 A) 1(점 B) 1(가중치)



백준 11675, 라벨링

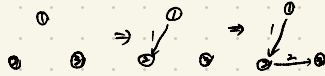
개념
벨만-포드 알고리즘

입력
3(점 A) 4(점 B)
1(점 A) 2(점 B) 4(가중치)



Step 1 모든 간선을 오름차순으로 정렬
1 2, 2 3, 1 3
1 2, 2 3, 1 3

Step 2 자중리가 적은 간선부터 연결
단, 이미 연결된 노드끼리 연결하면 안됨 (Union-Find)



Step 1 현재 지위를 나타내는 dist 배열을 무한대로 초기화

INF INF INF INF INF

1 2 3 4 5

Step 2 dist[시작점, 1]은 0으로 초기화

0 INF INF INF INF

1 2 3 4 5

Step 3 dist[i]가 1~2 자중리를 가진 간을 현재 dist[j]와 비교하여 작은 값으로 변경

이때, 기존 값보다 작을 때 dist 값을 위해 최소 값에 대한 1을 갱신해야 한다

0 2 INF INF INF

1 2 3 4 5

Step 4 최소 비용 갱신 0이 될 때까지 Step 3을 반복

0 2 3 7 INF

Step 1 dist 배열 (1부터 N 점까지 가는 시간)을 무한대로 초기화

INF INF INF

1 2 3

Step 2 dist[시작점, 1]은 0으로 초기화

0 INF INF

1 2 3

Step 3 전선 수 - 1 x 점들 수 만큼 기존 dist와 가중치를 더한 값을 작은 값으로 계속 업데이트

본 dist가 INF는 불가능하므로 갱신시켜 줌

0 INF INF → 0 4 INF → 0 4 3

Step 4 마지막까지 점들 수 만큼 업데이트를 하는데 기존 값보다 작아지면 바뀌어야 하는 값이 존재

0 4 3 → 0 0 3

값이 바뀌었으므로 해 사이클 존재

