

백준 1753, 최단 경로

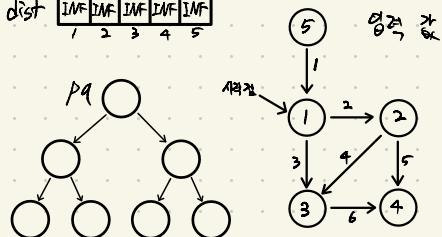
Step 1. 변수 초기화

dist: 시작점부터 최단 거리를 저장하는 배열

pq: 현재 경유지에서 최단 거리로 갈 수 있는

간선을 저장하는 priority queue (가중치 힙형)

dist	[INF INF INF INF INF]
	1 2 3 4 5

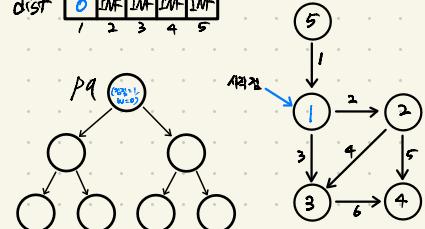


Step 2. 시작점 1의 dist를 0으로 초기화 pq에 간선 경로 (경정1, w=0) push

시작점 1의 dist를 0으로 초기화

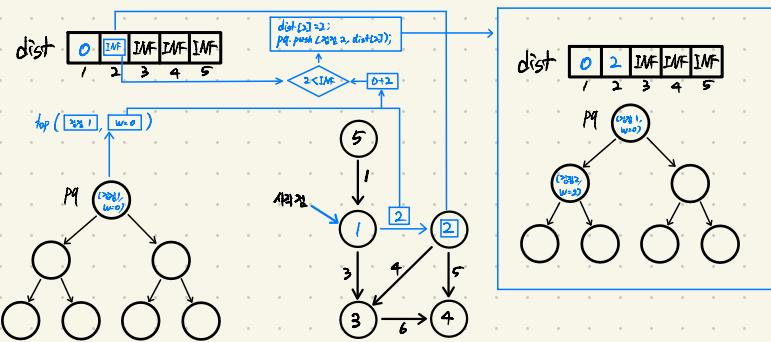
pq에 간선 경로 (경정1, w=0) push

dist	[0 INF INF INF INF]
	1 2 3 4 5



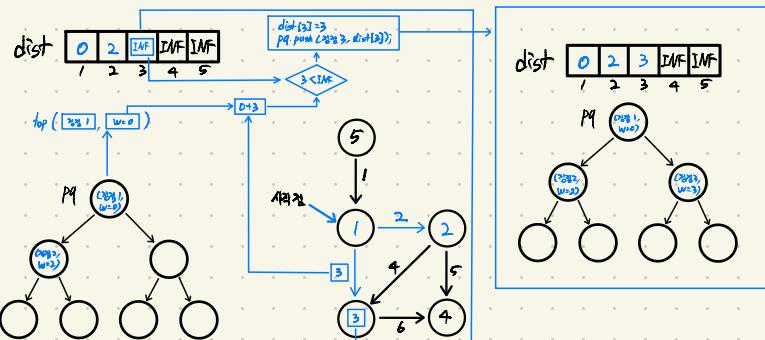
Step 3-1. pq의 top (경정1, w=0)의 경정1에서 경속했던 경정 2를 골라

마지막 경이 dist를 생성한다



Step 3-2. pq의 top (경정1, w=0)의 경정1에서 경속했던 경정 3을 골라

마지막 경이 dist를 생성한다



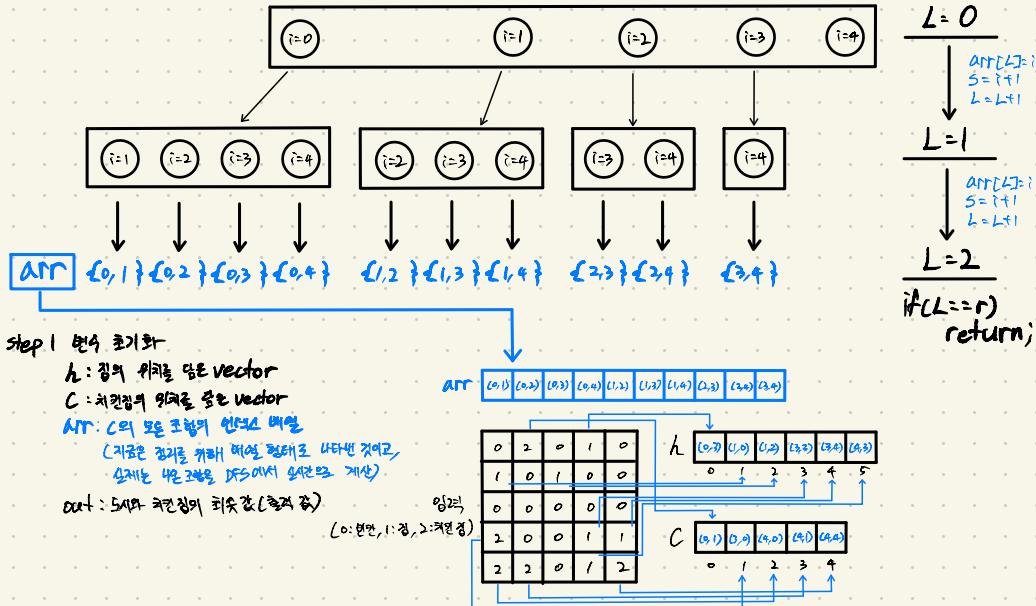
Step 4. pq가 모두 비어질 때까지 Step3 반복

Result: dist [0 2 3 7 INF]

백준 15686, 치킨 배달

모든 조합 구하기

DFS ($s=0, L=0$), $nCr (n=5, r=2)$



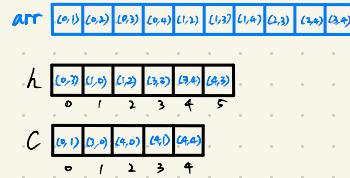
step 2 ARR 전체를 돌면서 최소값을 찾는다.

pseudo code

```

int out = INT_MAX; // 출력값 초기화
for(int i=0; i<ARR.size(); ) // 전체 대해 순회
{
    int sum = 0; // 현재 조합의 거리의 총
    // 정의 4
    for(int j=i; j<ARR.size(); )
    {
        int min_dist = INT_MAX; // 각 정마다 최초기기 값을 변수
        // 조합에 있는 치킨집 4
        for(int k=0; k<ARR[j].size(); )
        {
            pair<int, int> h-point = ARR[j]; // 정 위치
            pair<int, int> C-point = {ARR[j][k]} // 치킨 집 위치

            int dist = abs(h-point.first - C-point.first) +
                abs(h-point.second - C-point.second); // 거리 계산
            min_dist = min(min_dist, dist); // 최소 거리 갱신
        }
        sum += min_dist;
    }
    out = min(out, sum); // 출력 값 갱신
}
    
```

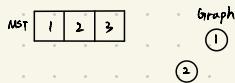


백준 1197, 최소 스패닝 트리

Step 1

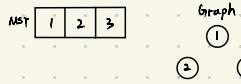
설명 : 원래 문제는 벡터
VEC : 원래 같은 정수들을 vector
SUM : 원래 같은 정수들

1	2	1	3	2	3
1		3		2	

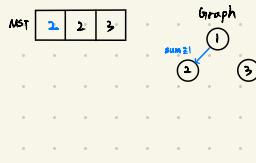
 (정수를 담아줄 예상되는 벡터) |

Step 2 MST 정수 가짐

1	2	2	3	1	3
1		2		3	



Step 3 MST VEC[0]에 정수 1을 더해 Union-Find 초기화
MST, SUM[0] = 1 → 2, 1, 3, 1, 3, 1



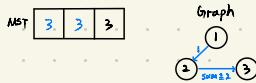
Union-Find 초기화

```
int find(int a) {
    if (MST[a] == a)
        return MST[a];
    else
        return MST[a] = find(MST[a]);
```

void Union(int a, int b)

```
{
    a = find(a);
    b = find(b);
    if (a != b) // a와 b가 다른 집합
        MST[a] = b; // a와 b를 합침
}
```

Step 4 MST VEC[1]에 정수 2를 더해 Union-Find 초기화



Step 4 MST VEC[2]에 정수 3를 더해 Union-Find 초기화



배운 75% 토마토

Step1. 변수 초기화
 day: 0으로 초기화
 box: 3x3 크기로 초기화
 q: 큐를 초기화하는 배열

box	<table border="1"> <tr><td>0</td><td>-1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>-1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> </table>	0	-1	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	-1	0	0	0	0																				
-1	0	0	0	0	0																				
0	0	0	0	0	0																				
0	0	0	0	0	1																				

day	<table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0																				
0	0	0	0	0	0																				
0	0	0	0	0	0																				
0	0	0	0	0	0																				

q	[]
---	-----

Step2. box 배열을 확인하면서 1인 곳의 좌표를 큐에 push

box	<table border="1"> <tr><td>0</td><td>-1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>-1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> </table>	0	-1	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	-1	0	0	0	0																				
-1	0	0	0	0	0																				
0	0	0	0	0	0																				
0	0	0	0	0	1																				

day	<table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0																				
0	0	0	0	0	0																				
0	0	0	0	0	0																				
0	0	0	0	0	0																				

q	(5,3)	[]
---	-------	-----

Step3. 큐에서 꺼낸 좌표를 기준으로 큐에 있는 모든 8방향을 체크
 box에 해당하는 day에 1을 더해 13번 반복
 큐에 있는 모든 좌표가 1이면 종료

box	<table border="1"> <tr><td>0</td><td>-1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>-1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table>	0	-1	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
0	-1	0	0	0	0																				
-1	0	0	0	0	0																				
0	0	0	0	0	1																				
0	0	0	0	1	1																				

day	<table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
0	0	0	0	0	0																				
0	0	0	0	0	0																				
0	0	0	0	0	1																				
0	0	0	0	1	1																				

q	(4,3)	(5,3)	[]
---	-------	-------	-----

q.front : (5,3)

Step4 Step5: 풀기 (99% 푸는 것과 차이가 있다)

box	<table border="1"> <tr><td>0</td><td>-1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>-1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> </table>	0	-1	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1
0	-1	0	0	0	0																				
-1	0	0	0	0	0																				
0	0	0	0	1	1																				
0	0	0	1	1	1																				

day	<table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>2</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>2</td><td>1</td><td>0</td></tr> </table>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	2	1	0
0	0	0	0	0	0																				
0	0	0	0	0	0																				
0	0	0	0	2	1																				
0	0	0	2	1	0																				

q	(2,3)	(3,3)	(4,3)	[]
---	-------	-------	-------	-----

q.front : (4,3)

Step6: 풀기 (99.9% 푸는 것과 차이가 있다)
 box에 1인 좌표를 큐에 넣고 day에 1을 더해 100번 반복
 큐에 있는 모든 좌표가 1이면 종료

box	<table border="1"> <tr><td>0</td><td>-1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>-1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table>	0	-1	1	1	1	1	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	-1	1	1	1	1																				
-1	1	1	1	1	1																				
1	1	1	1	1	1																				
1	1	1	1	1	1																				

day	<table border="1"> <tr><td>0</td><td>0</td><td>6</td><td>5</td><td>4</td><td>3</td></tr> <tr><td>0</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td></tr> <tr><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr> <tr><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr> </table>	0	0	6	5	4	3	0	6	5	4	3	2	6	5	4	3	2	1	5	4	3	2	1	0
0	0	6	5	4	3																				
0	6	5	4	3	2																				
6	5	4	3	2	1																				
5	4	3	2	1	0																				

q	[]
---	-----

백준 2178, 미로탈출

Step 1 맵과 초기화

- 이미지: 현재 칸의 상태를 같은 배열(0:벽, 1:통로, 2:출구, 3:목표)으로 표시하는 배열
- dist: 시작점에서 최단거리를 저장하는 배열(0으로 초기화)
- q: 절 수 있는 경로를 담은 Queue

arr	<table border="1"> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> </table>	1	0	1	1	1	1	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	0	1	1	dist	<table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	1	1																																														
1	0	1	0	1	0																																														
1	0	1	0	1	1																																														
1	1	1	0	1	1																																														
0	0	0	0	0	0																																														
0	0	0	0	0	0																																														
0	0	0	0	0	0																																														
0	0	0	0	0	0																																														

q []

Step 2 시작 좌표 L(1,1)을 큐에 push 한다.

해당 경로는 이미 처리되었으므로 연결된다.

arr	<table border="1"> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> </table>	1	0	1	1	1	1	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	0	1	1	dist	<table border="1"> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	1	1																																														
1	0	1	0	1	0																																														
1	0	1	0	1	1																																														
1	1	1	0	1	1																																														
1	0	0	0	0	0																																														
0	0	0	0	0	0																																														
0	0	0	0	0	0																																														
0	0	0	0	0	0																																														
		push	<table border="1"> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
1	0	0	0	0	0																																														
0	0	0	0	0	0																																														
0	0	0	0	0	0																																														
0	0	0	0	0	0																																														

q [1 0 0 0 0 0]

Step 3 Agent가 다음 칸을 살피는 경우 해당 큐에 push 한다.

여기서 큐에 넣을 때마다 dist[front] += 1을 한다.

마지막 front를 pop한다.

arr	<table border="1"> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> </table>	0	0	1	1	1	1	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	0	1	1	dist	<table border="1"> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1																																														
1	0	1	0	1	0																																														
1	0	1	0	1	1																																														
1	1	1	0	1	1																																														
1	0	0	0	0	0																																														
2	0	0	0	0	0																																														
0	0	0	0	0	0																																														
0	0	0	0	0	0																																														
		push	<table border="1"> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																								
1	0	0	0	0	0																																														
2	0	0	0	0	0																																														
0	0	0	0	0	0																																														
0	0	0	0	0	0																																														

q [1 0 0 0 0 0]

pop

Step 4 q가 모두 비어있을 때까지 Step 3을 반복 한다.

arr	<table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	dist	<table border="1"> <tr><td>1</td><td>0</td><td>9</td><td>10</td><td>11</td><td>12</td></tr> <tr><td>2</td><td>0</td><td>8</td><td>0</td><td>12</td><td>0</td></tr> <tr><td>3</td><td>0</td><td>7</td><td>0</td><td>13</td><td>14</td></tr> <tr><td>4</td><td>5</td><td>6</td><td>0</td><td>14</td><td>15</td></tr> </table>	1	0	9	10	11	12	2	0	8	0	12	0	3	0	7	0	13	14	4	5	6	0	14	15
0	0	0	0	0	0																																														
0	0	0	0	0	0																																														
0	0	0	0	0	0																																														
0	0	0	0	0	0																																														
1	0	9	10	11	12																																														
2	0	8	0	12	0																																														
3	0	7	0	13	14																																														
4	5	6	0	14	15																																														

q []

백준 11675, 탐색마인

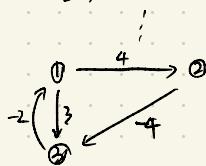
기회도

별안·포드 알고리즘

정역

3(정점 S) 4(전선 A)

1(정점 A) 2(정점 B) 4(가중치)



step 1 dist 배열 ([보더 쪽 정점 까지 가는 거리]을 무한 대로 초기화

INF	INF	INF
1	2	3

step 2 dist[1(정점 S), 1]은 0으로 초기화

0	INF	INF
1	2	3

step 3 간선수-1 × 정점 수 만큼 거리를 떠나와 가중치를 더한 값을 차운 것으로 계속 업데이트
문, dist과 INF는 동시에 업데이트 계산하여 처리

0	INF	INF
1	2	3

 \rightarrow

0	4	INF
1	2	3

 \Rightarrow

0	4	3
1	2	3

step 4 마지막으로 정점 수만큼 업데이트 하는데 기본값보다 작아져면 사이클이 존재

0	4	3
1	0	3

같이 출력적으로 흑위 사이클 존재

