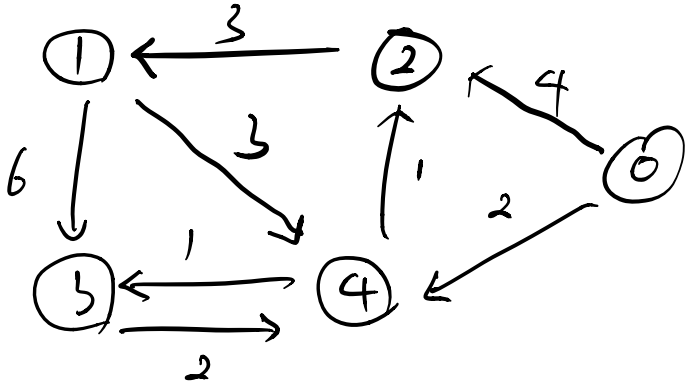
다익스트라 알고리즘

- 하나의 정점에서 다른 모든 정점까지의 최단경로
- ★ 모든 양의 간선을 가져야함

연결되어있는 정점들을 추가해가며, 최단거리 갱신

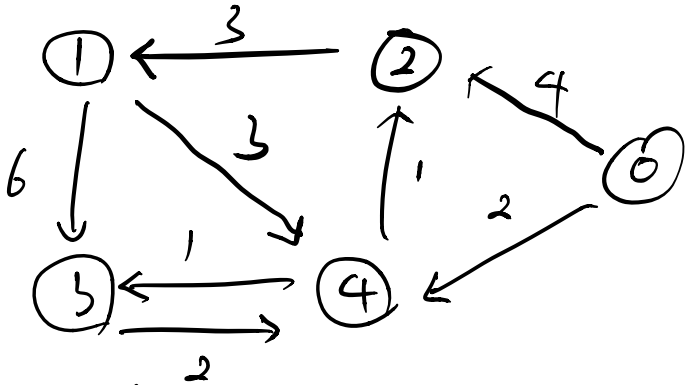


0번에서 시작!

node	0	1	2	3	4	init step
거리	0	INF	INF	INF	INF	
node	0	1	2	3	4	1 step
거리	0	INF	4	INF	2	
node	0	1	2	3	4	2step
거리	0	INF	3	3	2	
node	0	1	2	3	4	3step
거리	0	INF	3	3	2	
node	1	2	3	4	5	4step
거리	0	6	3	3	2	

☆☆ priority queue를 이용한 다익스트라 알고리즘

priority queue에 dist(간선의 value)를 기준으로 넣어줘라.
→ priority queue를 통해 가장 작은 값을 빼내라



0번에서 시작!
pq가 비게 될 때까지 1, 2 step 반복

S	d	v
0	2	4
0	4	2

→ priority queue에 넣어줌.

node	0	1	2	3	4
거리	0	INF	4	INF	2

↓ pq.delete

(0 4 2)가 나옴 pq 0 2 4

↓

4번 node와 연결된 값들 중,
node[4] + 4번 node와 i번째 node 사이거리 < node[i] 일경우
node[i]를 update 해주고 priority queue에 push

pq 4 3 3
4 2 3
0 2 4

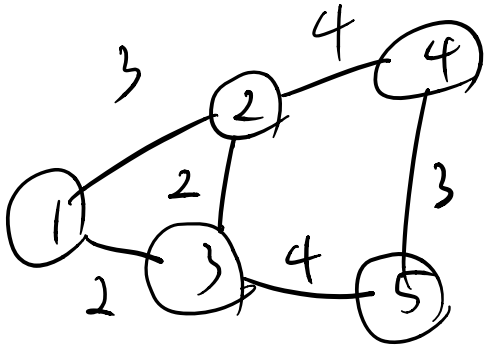
1 step

2 step

간선 예약 step

n1	n2	value
1	2	3
1	3	2
2	3	2
2	4	4
3	5	4
4	5	3

1. 간선예약은 cycle이 있었기 하지만 cycle 가능
2. 간선예약은 이번 과제스트라 사용



예제 시작

pq
Step 1 1 3 2
1 2 3

node	1	2	3	4	5
dist	0	3	2	INF	INF

↓ pq delete (1, 3, 2)

pq
step 2 1 2 3
3 5 6

node	1	2	3	4	5
dist	0	3	2	INF	6

↓ pq delete (1, 2, 3)

pq
step 3 3 5 6
2 4 7

node	1	2	3	4	5
dist	0	3	2	7	6

↓ pq " (3, 5, 6) " "
step 4 2 4 7

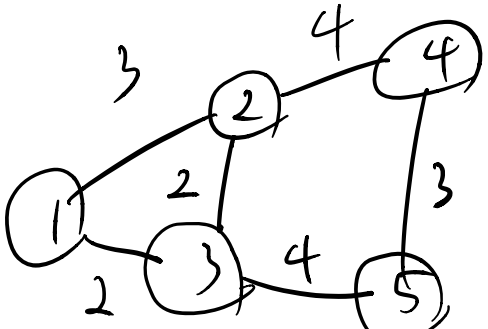
↓ " (2, 4, 7) " "
step 5 empty

칼빈 늘려 step

1. 칼빈늘려는 짝수_홀수 따져봐

n1	n2	value
1	2	3
1	3	2
2	3	2
2	4	4
3	5	4
4	5	3

2.5



pq

Step 1. 1 3 2 odd
1 2 3 odd

odd 1 2 3 4 5
INF 1.5 1 INF INF
even INF INF INF INF INF

↓ pq del (1, 3, 2, odd)

Step 2. pq
1 2 3 odd
3 2 5 even
3 5 9 even

odd 1 2 3 4 5
INF 1.5 1 INF INF
even INF 1+4 INF INF 1+8

↓ pq del (1, 2, 3, odd)

Step 3 3 2.5 e
2 3 5.5 e
3 5 9 e
2 4 9.5 e

odd 1 2 3 4 5
2 1.5 1 2 2
even 5 5.5 9.5 9

↓ pq del (3, 2, 5, e)

Step 4 2 3 9.5 e
2 1 6.5 0
2 4 9 e 0
3 5 9 e
2 4 9.5 e

odd 1 2 3 4 5
6.5 1.5 1 9 2
even 5 5.5 9.5 9

14
12