Djikstra algorithm

발표자 김윤재

What is djikstra algorithm?

다익스트라 알고리즘은 dynamic programming을 이용한 대표적인 최단 경로 탐색 알고리즘입니다.

GPS software에서 굉장히 많이 사용되는 알고리즘이며, 음의 간선을 포함할 수 없습니다. 그러므로 다익스트라는 현실 세계에 사용하기 매우 적합한 알고리즘 중 하나입니다.

다익스트라 프로그래밍 과정

1. 출발 노드를 지정합니다.
2. 출발 노드를 기준으로 각노드의 최솟값을 저장합니다.
3. 방문하지 않은 노드 중 가장 비용이 적은 노드를 선택합니다.
4. 해당 노드를 거쳐서 특정한 노드로 가는 경우 최소 비용을 갱신합니다.
5. 위 과정에서 3~4번을 반복합니다.
   다익스트라 알고리즘은 시간복잡도 O(N^2)가 나오지만,
   잘 구현된 priority queue를 사용하면 O(NlogN)까지 줄일 수

출발 노드를 지정한다.

노드가 5개라고 가정했을 때,

1차원 배열을 선언합니다.

그리고 시작 노드에는 값 0을, 다른 노드에는 INF=무한값을 넣어줍니다.

INF 값은 버퍼 오버플로우가 일어날 수도 있으니, 문제에 따라 다르게 배정해줍니다.

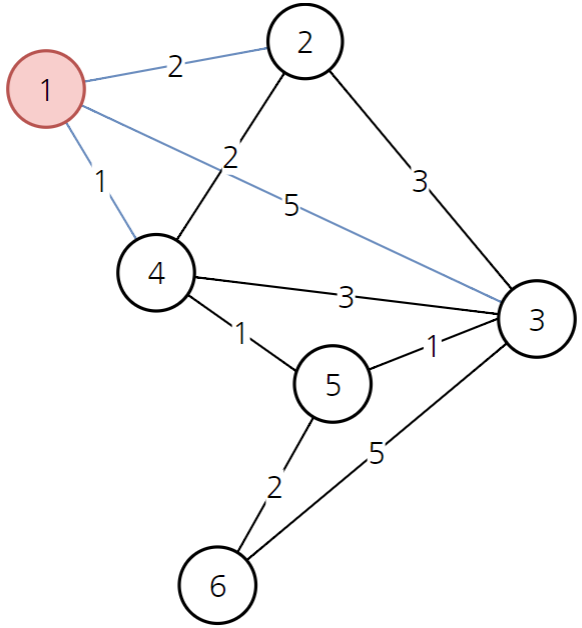
출발 노드를 기준으로, 각 노드의 최솟값을 지정한다.

출발 노드에 연결되어 있는 모든 값을 탐색하여, 연결되어 있는 배열에 그 값을 대입한다.

이 사진같은 경우에는, 연결되어있는 2,3,4 노드에, 각각의 값을 대입하면 된다.

각각의 값을 대입하는 과정에서 , 1번 노드 갖고 있는 값은, 무조건 INF보다 작기 때문

값이 대입되는 과정을 거친다.

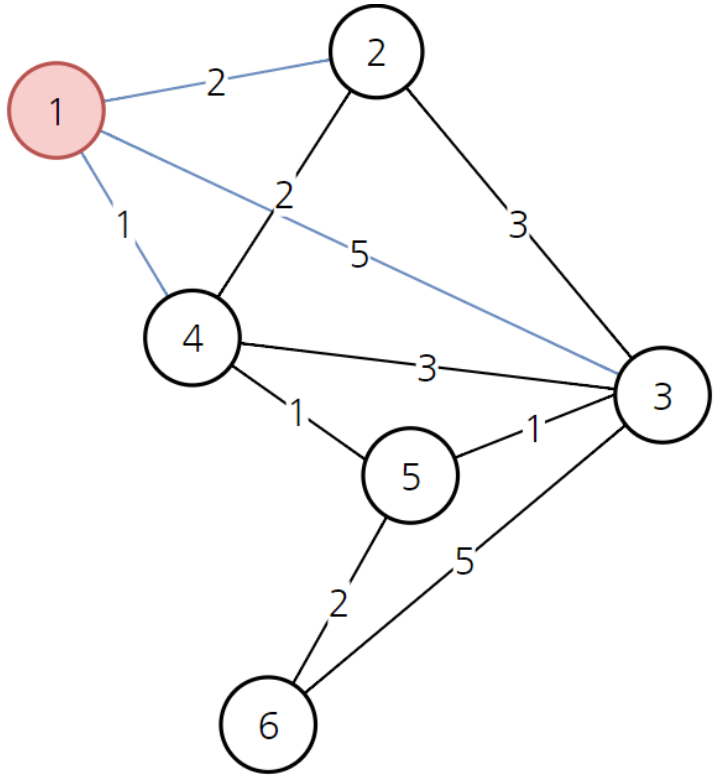


**해당 노드를 거쳐서 특정한 노드로 간값이 최솟값일경우,
노드의 최솟값을 갱신한다.**

이 사진의 경우에는, 1->2->5/1->3/1->4->5의 순서로 접근하는 세 가지 방법이 있는데, 이미 첫 번째 과정에서 3번 노드에 5라는 값이 들어가버렸다.

다익스트라 알고리즘은 4에서 3의 노드에 접근할때, 나의 값 4와 이미 있는 값 5를 비교하여, 최솟값인 4로 3번 노드의 값을 바꾼다.

그렇지만, 1->4->5->3으로 가면, 3이라는 최솟값을 얻을 수 있다.
그래서 4보다 3에 먼저 접근한다.



아래 과정을 반복하면, 모든 노드에 최솟값이 들어가게 된다.

Q : 그러면 간선의 값이 음수인 경우, 무한 순환에 빠지게 되지 않나요?

A : YES. 그래서 그럴 때는 다익스트라보다 비효율적이지만 음수의 간선을 구분할 수 있는 벨만 포드 알고리즘을 사용한다.

O(N^3)의 시간 복잡도를 가지고 있기 때문에, 음수의 간선이 있는게 확실할 때만 사용한다.

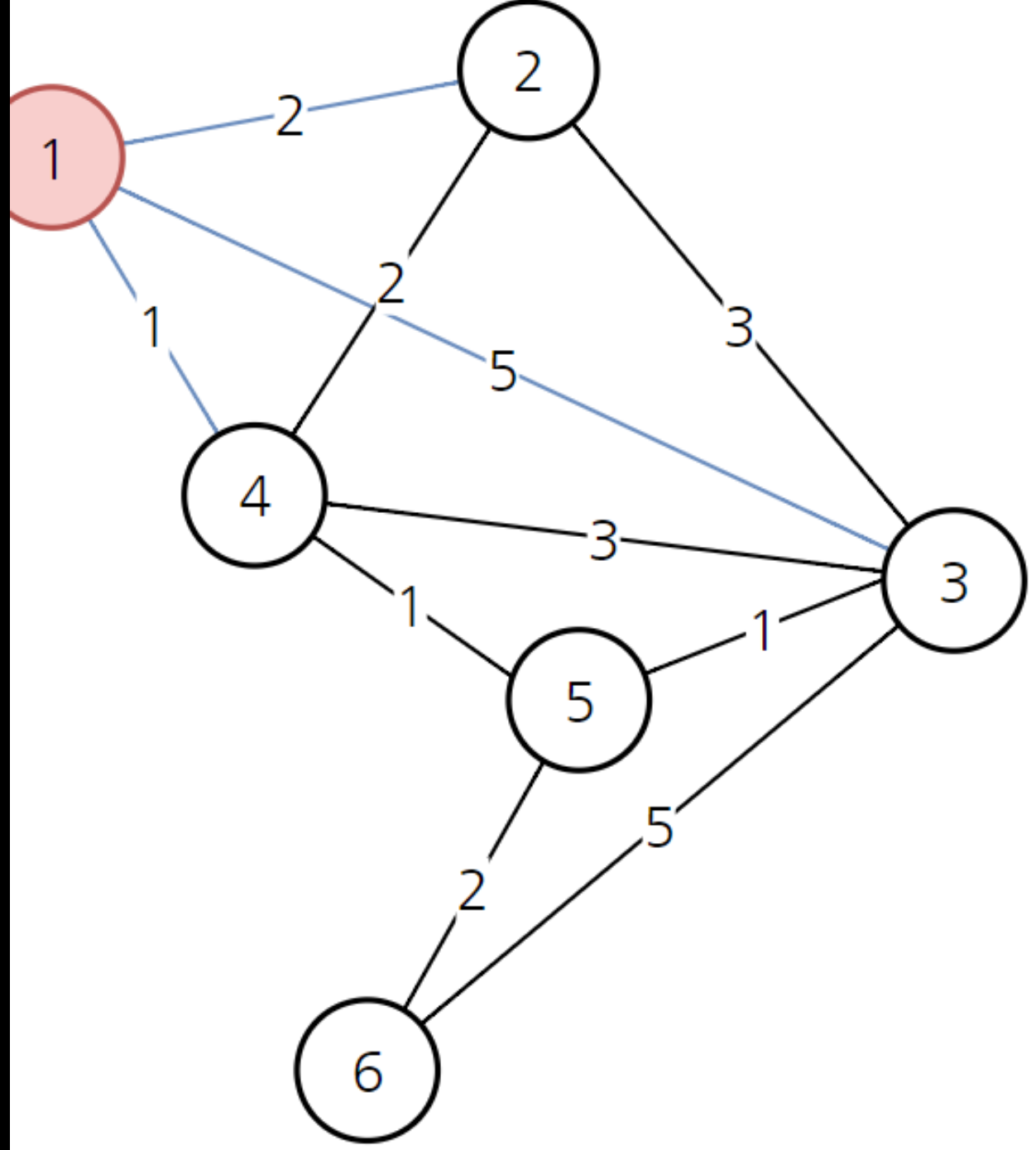
Priority queue를 쓰면, 왜 NlogN일까?

Bfs 탐색 과정을 거친다는 가정하에, pq를 사용하면 일어나는 일을 큐로 그려본다면.

(우측 프로트)

1

Dest:3 Value:5	Dest:2 Value:2	Dest:4 Value:1



2순회 :

Dest : 3 Value : 4	Dest:5 Value:2	Dest:3 Value:5	Dest:2 Value:2

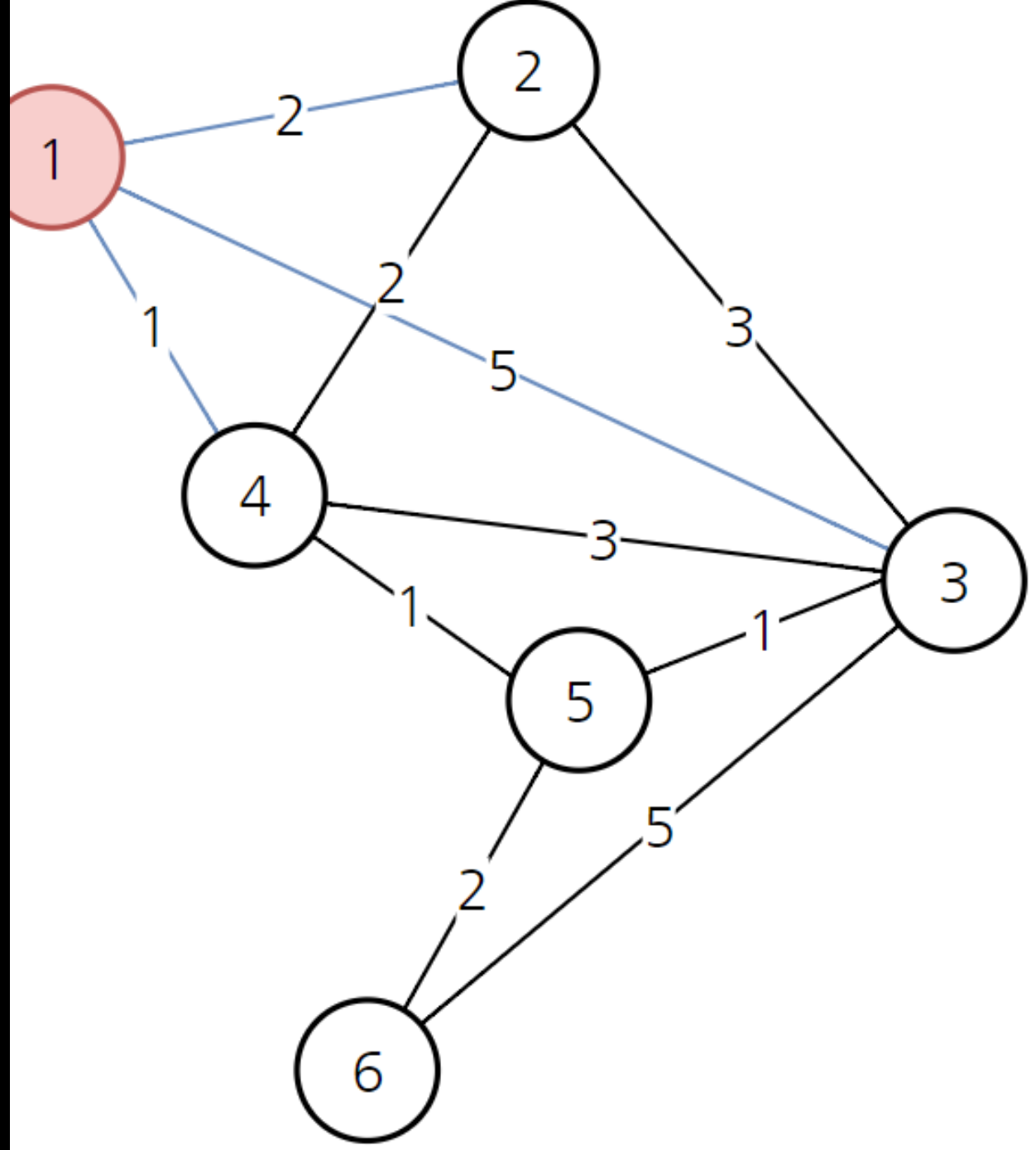
3순회 :

	Dest : 3 Value : 5	Dest : 3 Value : 4	Dest:5 Value:2

4순회 :

Dest : 3 Value : 5	Dest:6 Value:4	Dest : 3 Value : 4	Dest:3 Value:3

Queue에 더 낮은 가중치의 값이 들어올
때마다, 갱신을 해주기 때문에
우선순위 큐를 쓰지 않은 것보다 빠른



감 사 합 니 다.