# 자료 구조 스터디 07

2021-2 KCA

Dijkstra

본 ppt의 자료는 Pt.J님의 자료와 김성열 교수님의 강의 및 여러 블로그를 참고하였음을 밝힙니다.

# 그래프의 표기

- V : Vertex (꼭짓점)
- E : Edge (간선)

- G(V, E)

# 다익스트라 알고리즘 (Dijkstra)

- 그리디 + 다이나믹 프로그래밍
- 음의 가중치가 없는 그래프에만 적용 가능
- 한 노드에서 다른 모든 노드까지의 최단 거리를 구할 때 사용
- 그래프에서 노드 탐색 순서는 BFS(너비 우선)를 따른다.

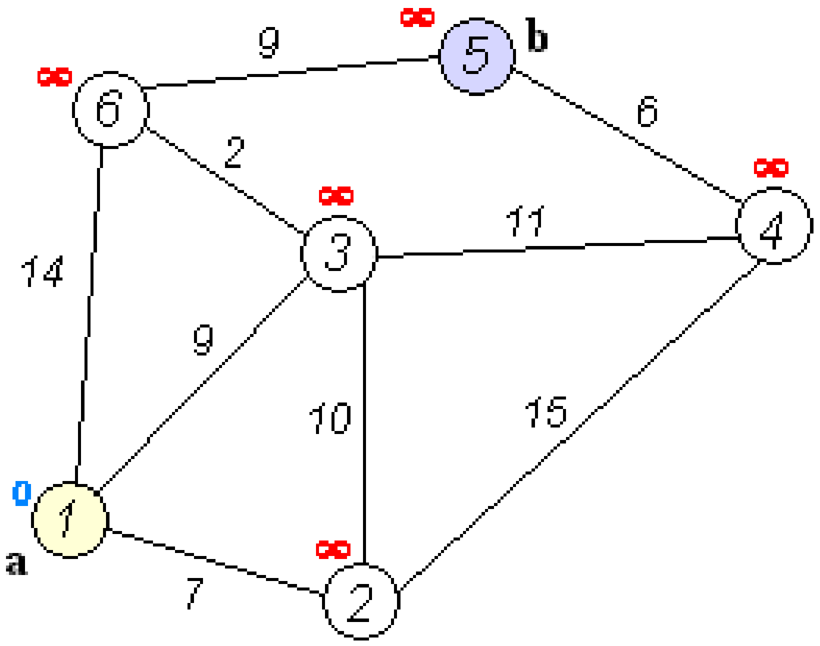
# 일반적 구현

1. 너비 우선으로 탐색하면서,

2. 방문하지 않은 노드 중 가장 비용이 적은 노드를 선택 (그리디)

3. 구해진 가장 적은 비용을 이전 비용과 합산하여 저장 (DP)

- 최악의 경우, 모든 노드를 방문해서 비용을 확인하므로 V, 이를 꼭짓점 개수만큼 반복하므로 $O(V^2)$
- 엄밀히는 간선 저장 시간(보통 배열이나 연결리스트로 저장)까지 $O(V^2 + E)$



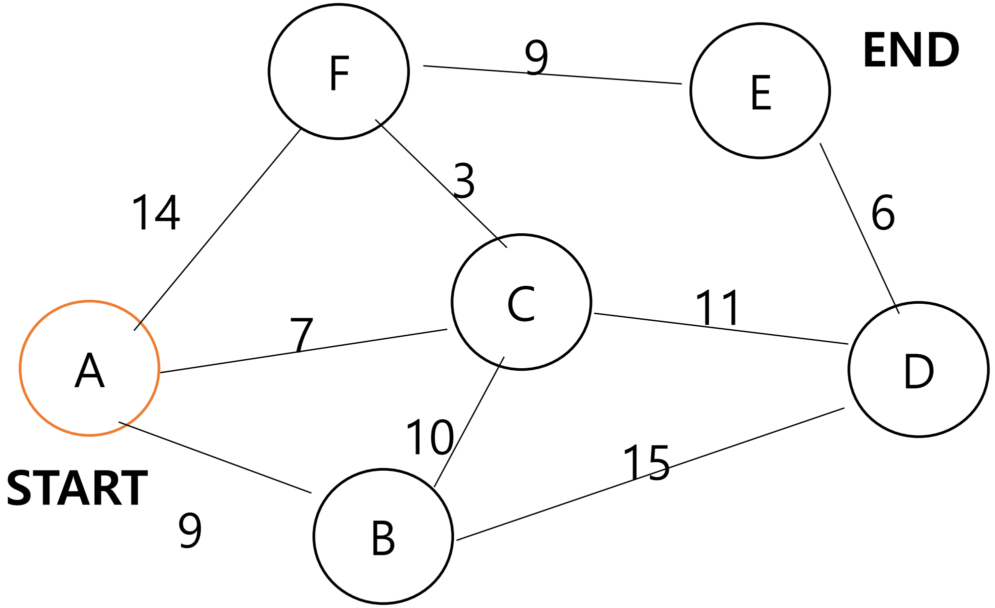

https://ko.wikipedia.org/wiki/%EB%8D%B0%EC%9D%B4%ED%81%AC%EC%8A%A4%ED%8A%B8%EB%9D%BC_%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98

- 최단 경로의 부분 경로도 최단 경로이다.

# 우선순위 큐 이용한 구현

- 최소 비용을 갖는 노드를 선택하되, 우선순위에 따라 선택한다.
- O((E + V) log V) => O(ElogV)

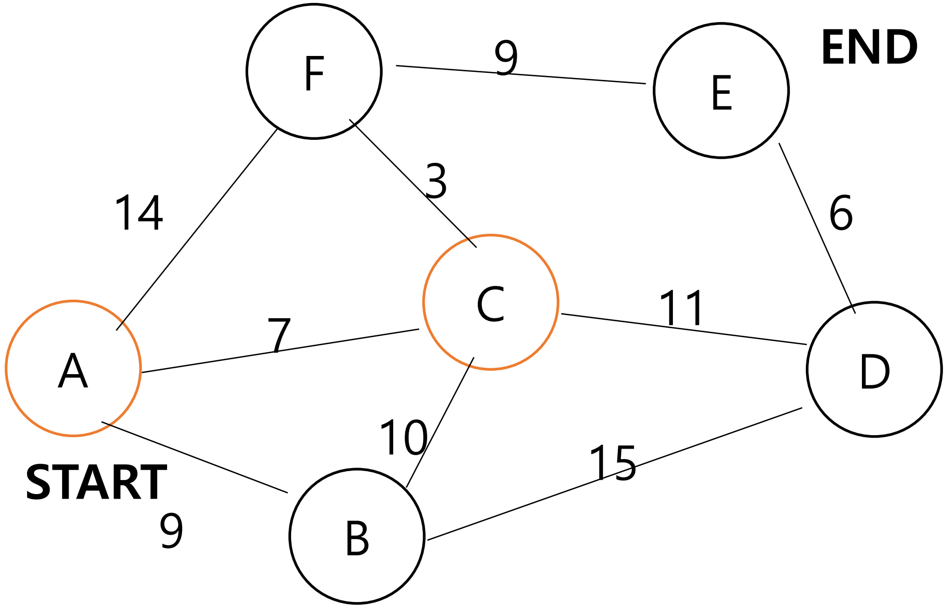
- Min Heap 을 이용한 우선순위 큐에,
  BFS를 진행할 때처럼 원소를 집어넣고 빼면서 탐색해 나간다.




(A, 0)

(B, 9), (C, 7), (F, 14)

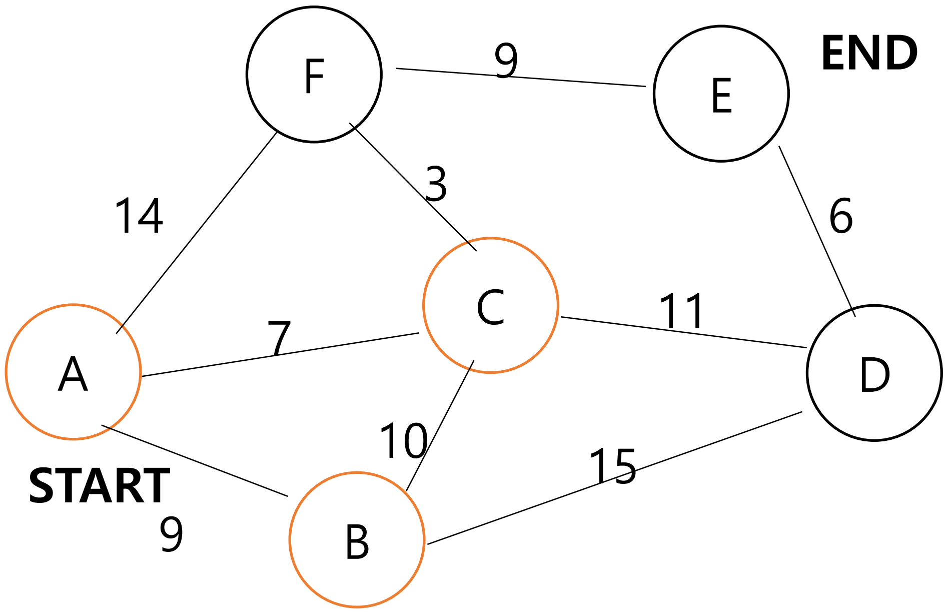
>> (C, 7) 먼저 Pop




(B, 9), (F, 14)

>> (B, 17), (D, 18), (F, 10)
비교 후 작다면(우선순위가 크다면) 갱신
전부 큐에 넣어놓은 다음, 꺼냈을 때 이미 방문한 곳이면
무시하는 것이 더 편하다.

(B, 9), (F, 10), (D, 18)

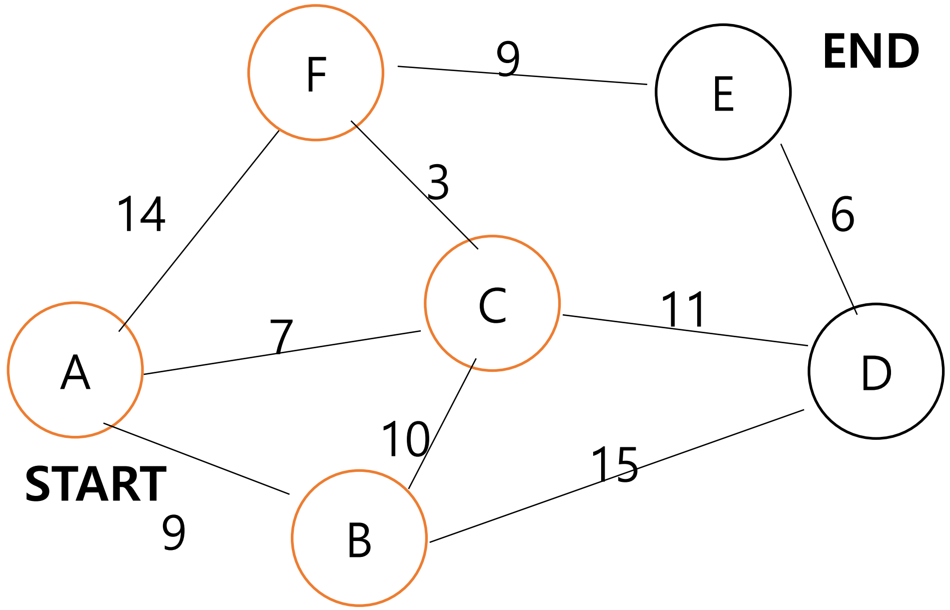



\>> (B, 9) Pop

(F, 10), (D, 18)

\>> (D, 24)와 비교

(F, 10), (D, 18)

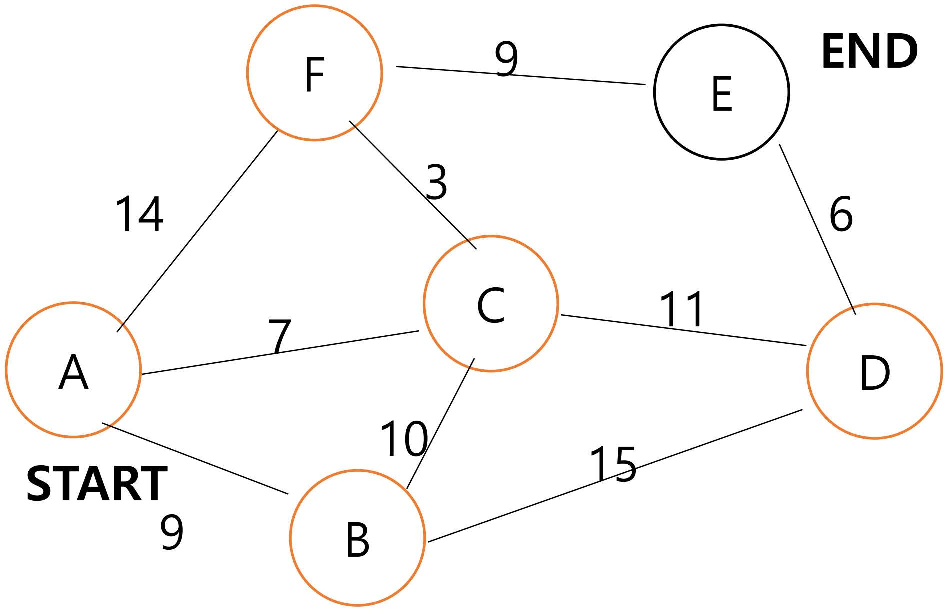



\>> (F, 10) Pop

(D, 18)

\>> (E, 19)와 비교

(D, 18), (E, 19)



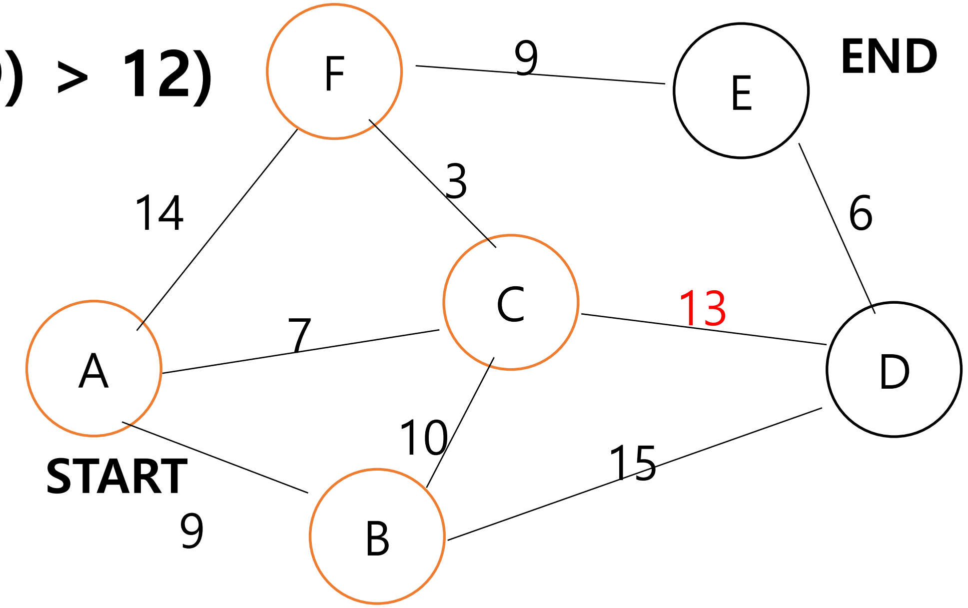

>> (D, 18) Pop

(E, 19)

>> (E 24)와 비교

(E, 19)

If(E(C, D) > 12)




(E, 19), (D, 20)

>> (E, 19) Pop
목적지에 도달했으므로 바로 종료 가능

=> 우선순위 큐를 쓰면 더 효율적이다!!

# 정확성 증명

- 귀류법 이용
- 노드 S 출발, E 도착.
- 명제 : 다익스트라를 쓰면(근접한 노드 중 가장 적은 가중치를 선택하고, 가중치를 계속 갱신하면서 탐색하면) E에 도착했을 때 최단거리라는 것이 보장된다.

- S~E가 최단거리가 아니라고 할 때,
  그렇다면 다른 노드를 지나는 경로가 최단거리가 될 것이다.
- 그러나, 다익스트라 조건에 의해 근접한 노드들 중 가장 적은 가중치를 계속 선택해야 하므로,
  중간 과정에서 가장 적은 가중치를 가지지 않은 다른 노드를 선택한다면 최단거리가 아니게 된다.
- 따라서 "다른 노드를 지나는 경로가 최단거리가 된다"와 모순되므로 귀류법으로 증명 완료.

# 시간복잡도

- 그래프에서 최대 간선(Edge)의 개수 : 임의의 두 노드를 뽑는 모든 경우의 수
  = ${}_{V}C_{2}$ = V(V-1)/2 => O(E) = O($V^2$)

- 우선순위 큐에 모든 간선의 개수 만큼 삽입 연산이 일어남. => O(ElogE) = O(2E*logV)

- 우선순위 큐에 모든 노드의 개수 만큼 추출 연산이 일어남. => O(VlogV)

총 시간복잡도 : O( 2E*logV + VlogV ) = O((2E+V)logV) == O(ElogV)

단, E < V인 경우도 간혹 있다! (단절 그래프, 트리 형태 DAG 등)

# 여러 최단 경로 알고리즘은 알고리즘 시간에...

- 다익스트라
- A*
- 벨먼-포드
- 플루이드-워셜