6. 최단경로 알고리즘

- 한지점에서 다른 한지점까지 최단경로

- 한지점에서 모든지점까지 최단경로

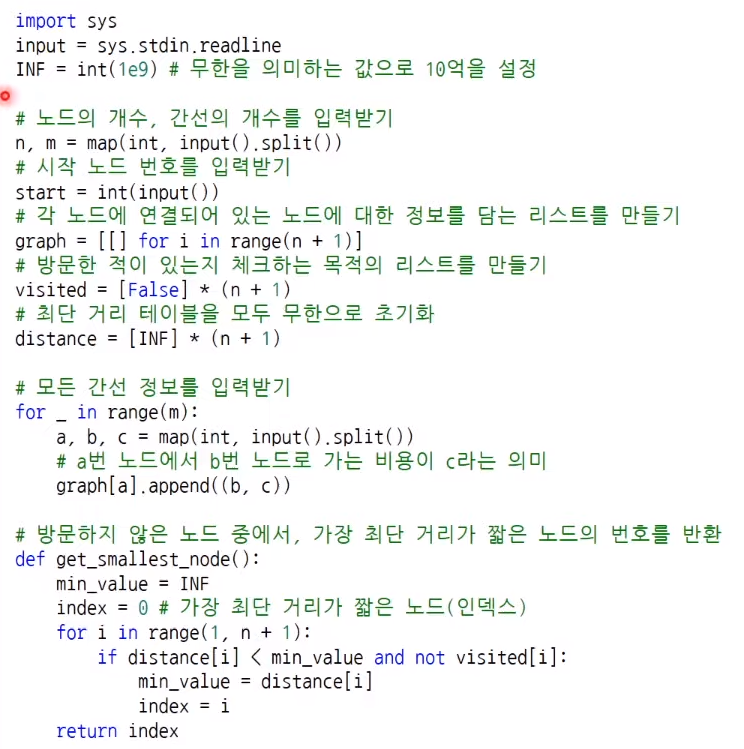
Dijkstra 알고리즘

* 특정노드에서 출발하여 다른 모든 노드로 가는 최단경로
* 그리디로 분류

1. 출발노드 설정
2. 최단거리 테이블 초기화
3. 방문하지 않은 노드 중에서 가장짧은 노드 선택
4. 해당 노드를 거쳐 다른노드로 가는 비용을 계싼하여 최단거리 테이블 갱신
5. 3,4반복

처리과정에서 더 짧은 경로 찾으면 “이제 이게 제일 짧은 경로야” 라고 갱신

단계를 거치며 한번 처리된 노드의 최단거리 고정

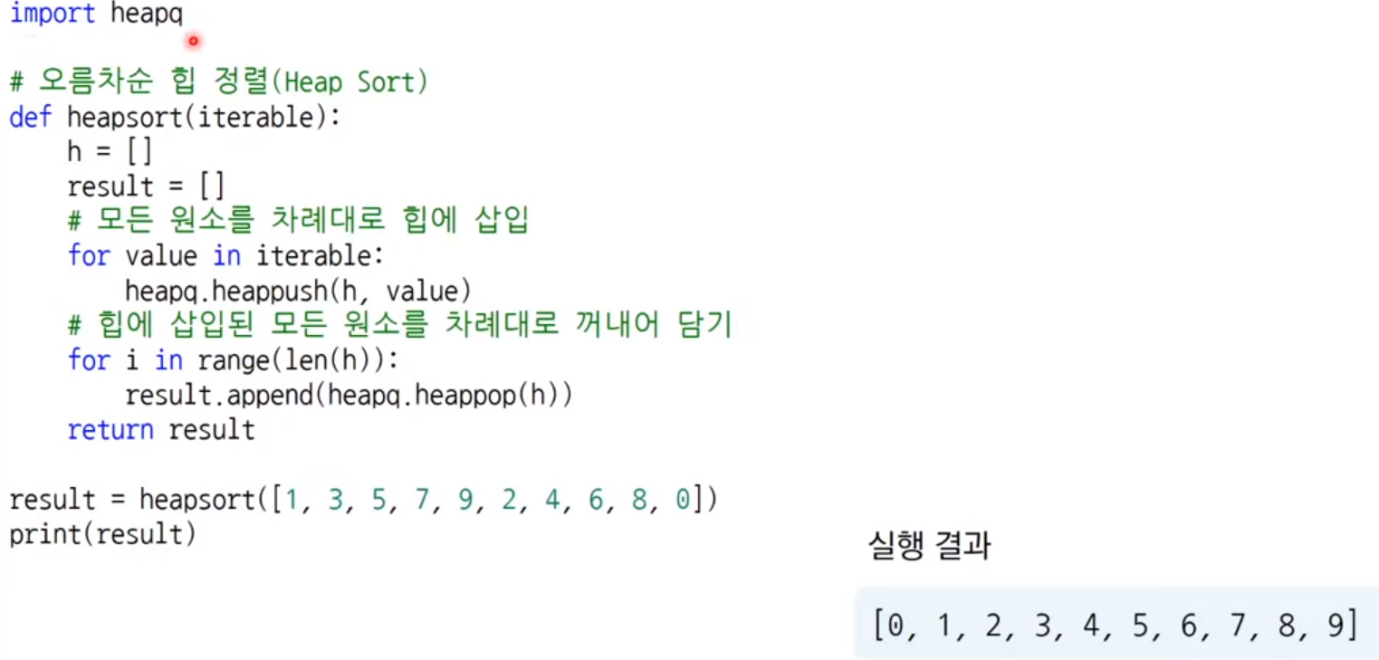


노드의개수 5000개이하라면 가능

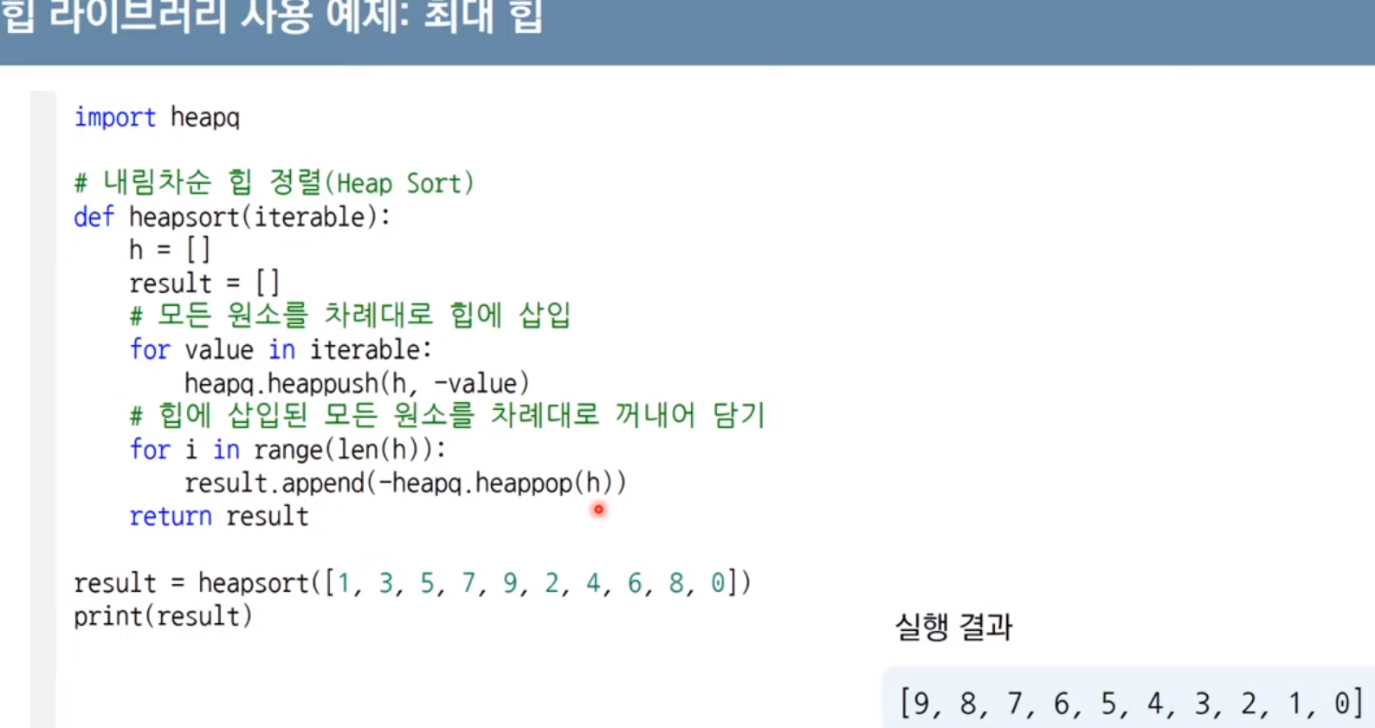
만번 이상이면?

우선순위큐

* 우선순위가 가장 높은 데이터를 가장 먼저 삭제
* 구현 위해 힙 사용- 최소힙(Min Heap), 최대힙(Max Heap)

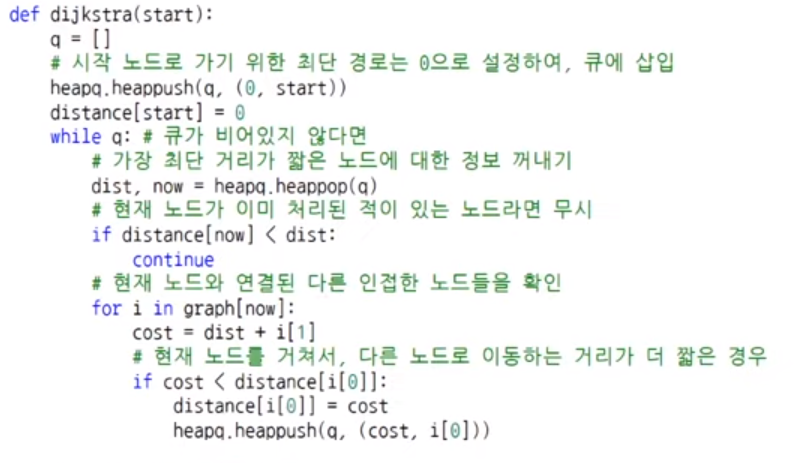


파이썬은 기본적으로 최소힙, 힙은 꺼낼 때 자동적으로 정렬이 된다.



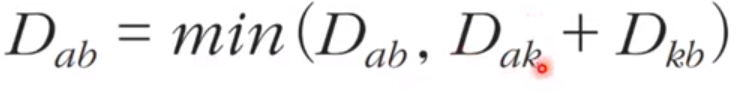
최대힙은 값의 부호 바꿔서 넣고 빼야함

우선순위큐 구현

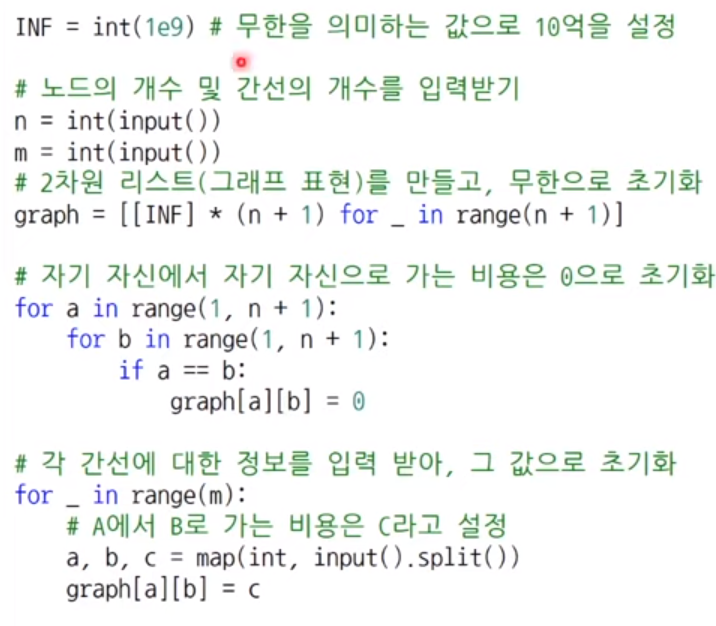


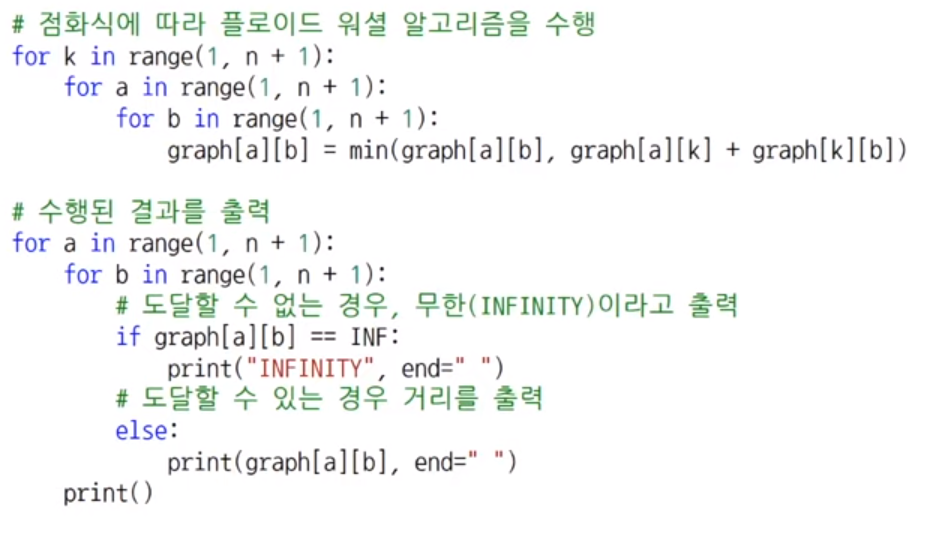
최대 노드의 개수만큼만 반복됨

플로이드 워셜 알고리즘

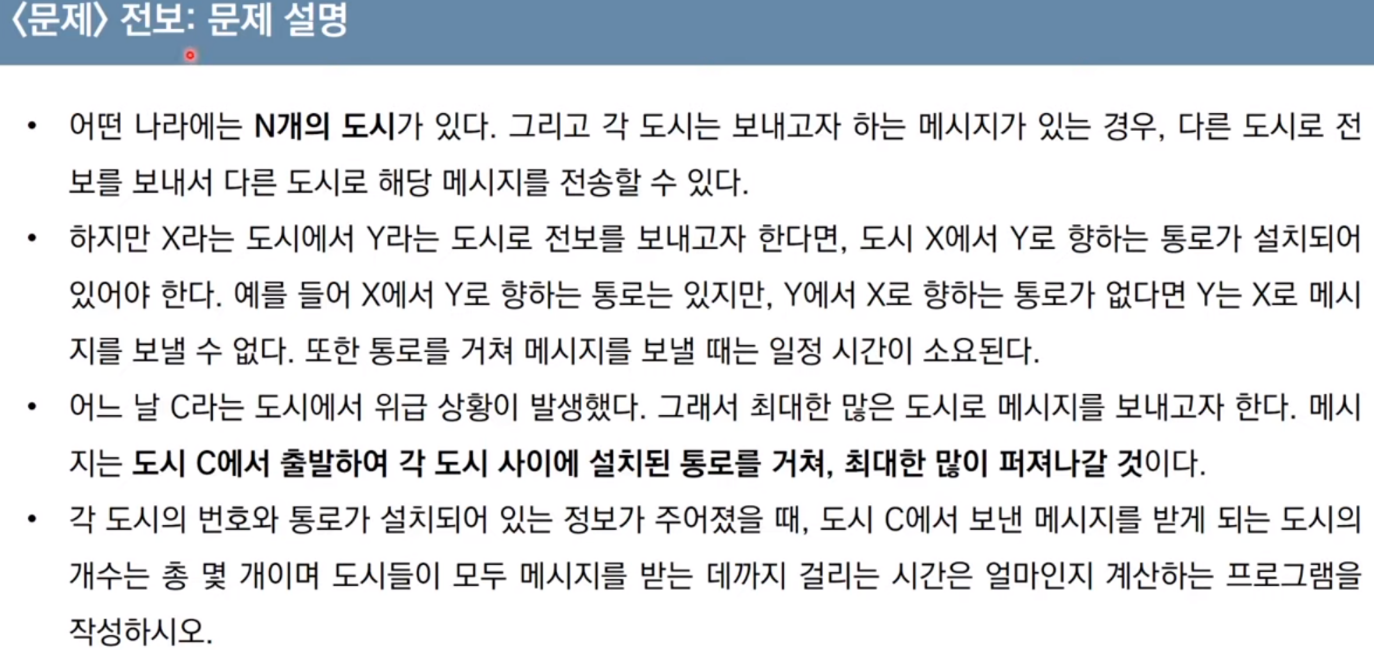
* 모든 노드에서 다른 모든 노드까지 최단경로
* 2차원테이블 이용
* 다이내믹 유형
* 노드의 개수가 적은 경우 사용
* K를 거쳐 가는 경우가 더 짧은지 확인
* 

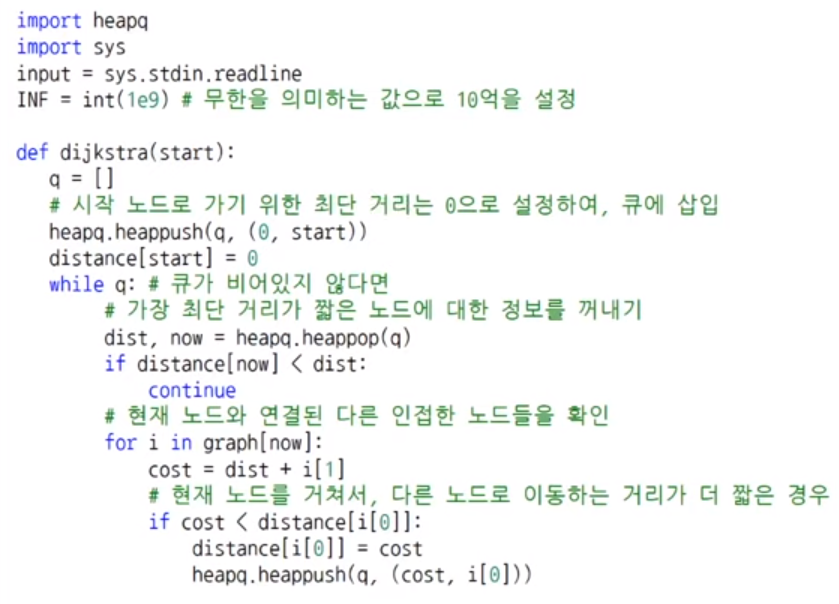
구현

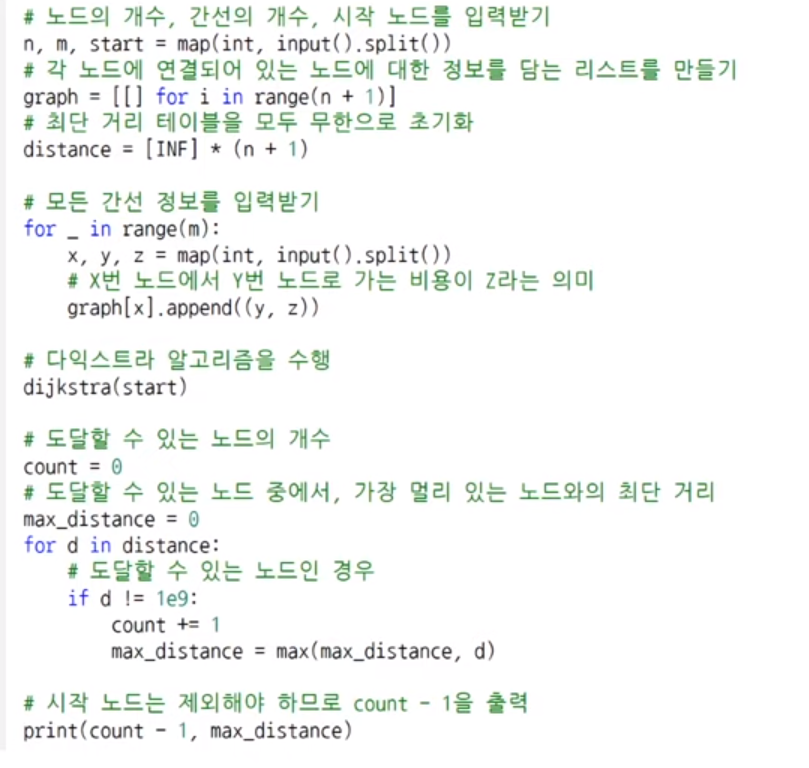


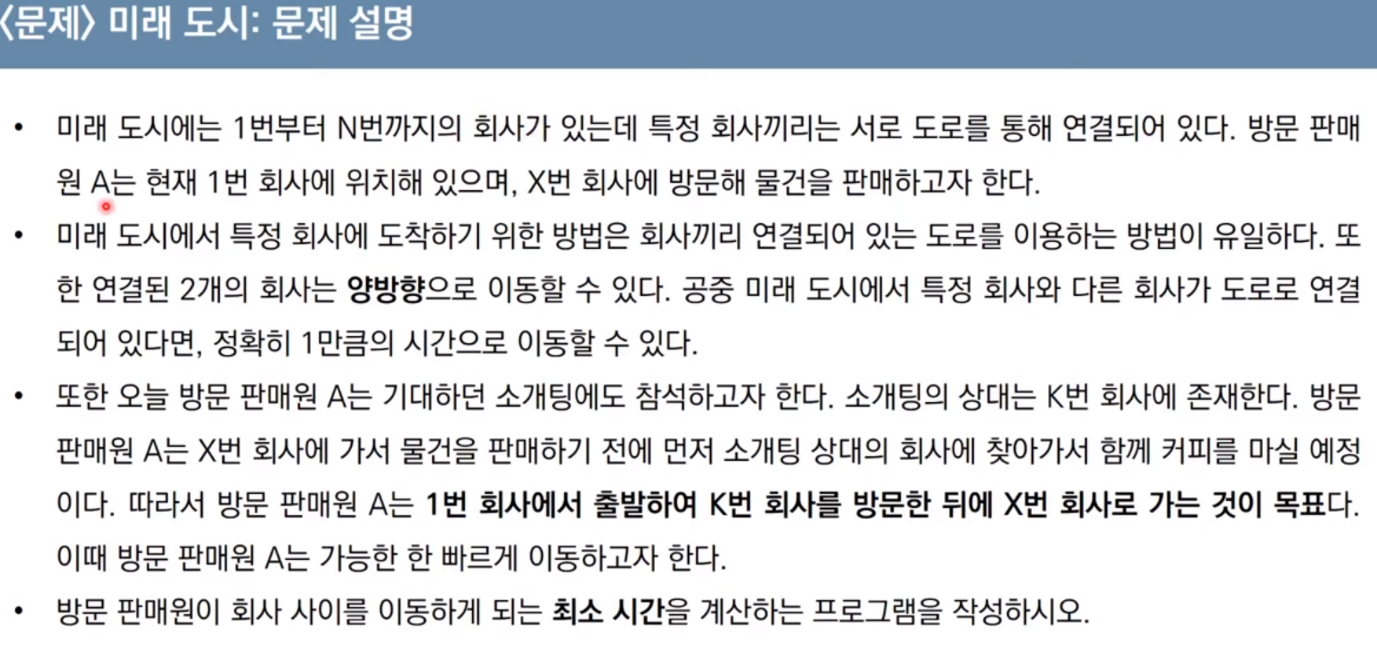


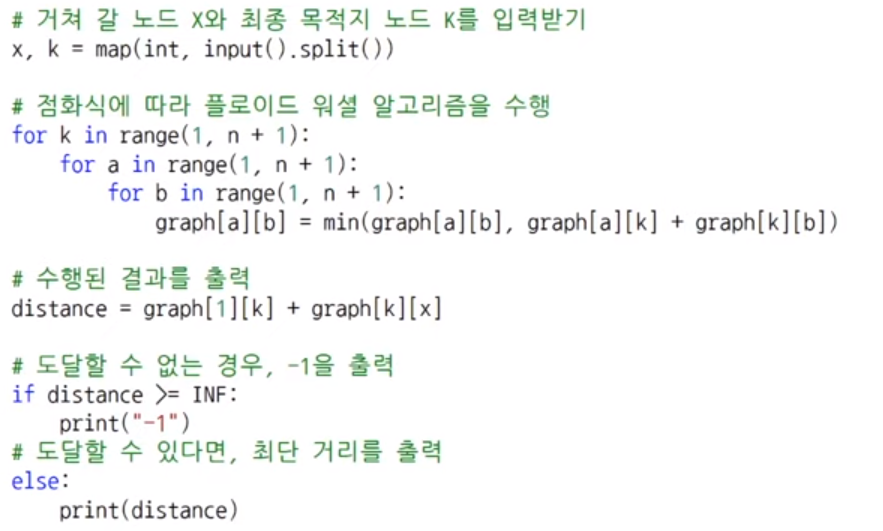
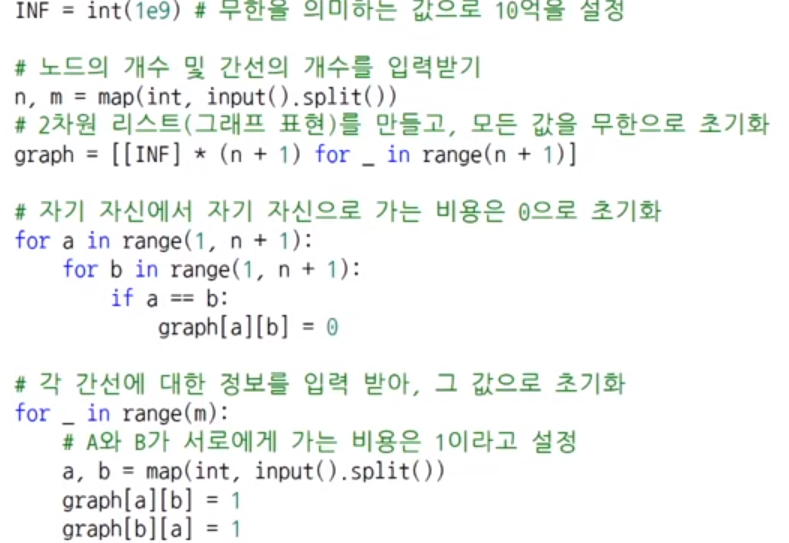
500개 이상의 노드는 처리하기 어려움











벨만 포드알고리즘

* 음수 간선 있을 때사용 가능

