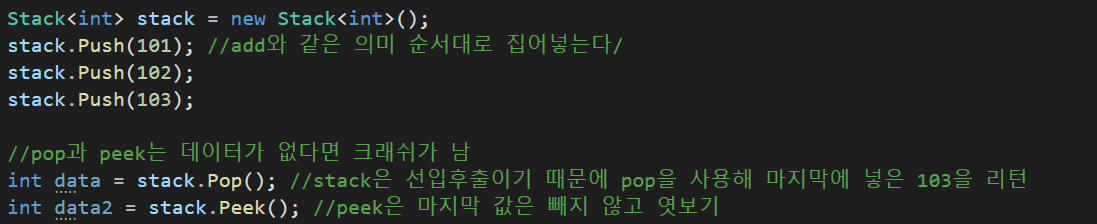
1. 스텍 & 큐

* 선형 자료구조
* 스택 : 후입선출(LIFO)



* 큐 : 선입선출(FIFO)

텍스트이(가) 표시된 사진

자동 생성된 설명

* 스택은 후입선출, 큐는 선입선출인데, 연결 리스트는 둘다 가능하다. 그럼 왜 연결 리스트를 쓰지 않고 스택이나 큐를 사용하는가? 추상적으로 사용할 때 좋기 때문에다. 또한 연결 리스트를 구현할 때 서로 이어주는 노드를 만드는 것이 부담이 될 수 있다.

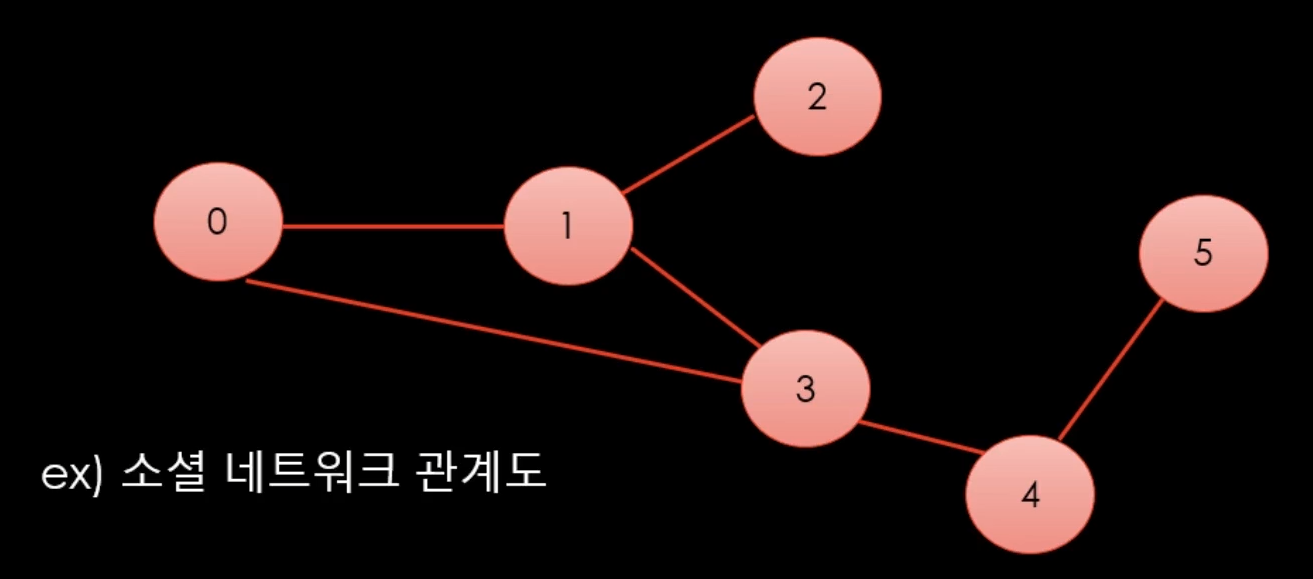
1. 그래프 이론

* 현실 세계의 사물이나 추상적인 개념간의 연결 관계를 표현

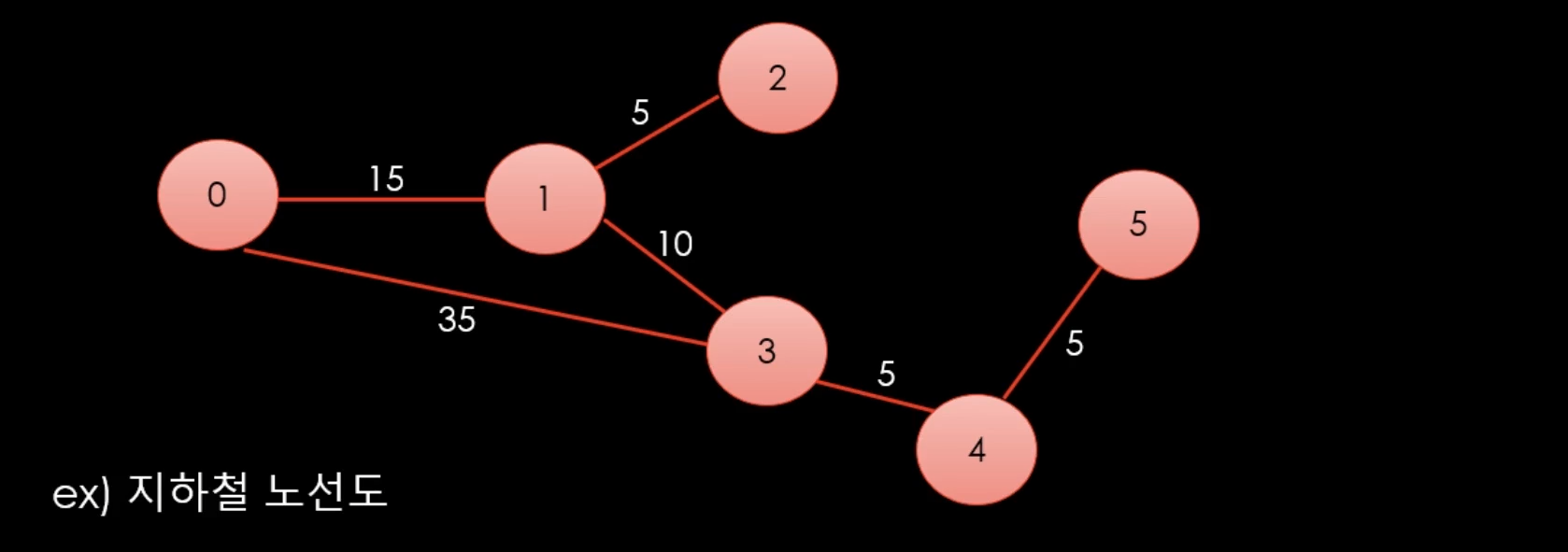
텍스트이(가) 표시된 사진

자동 생성된 설명

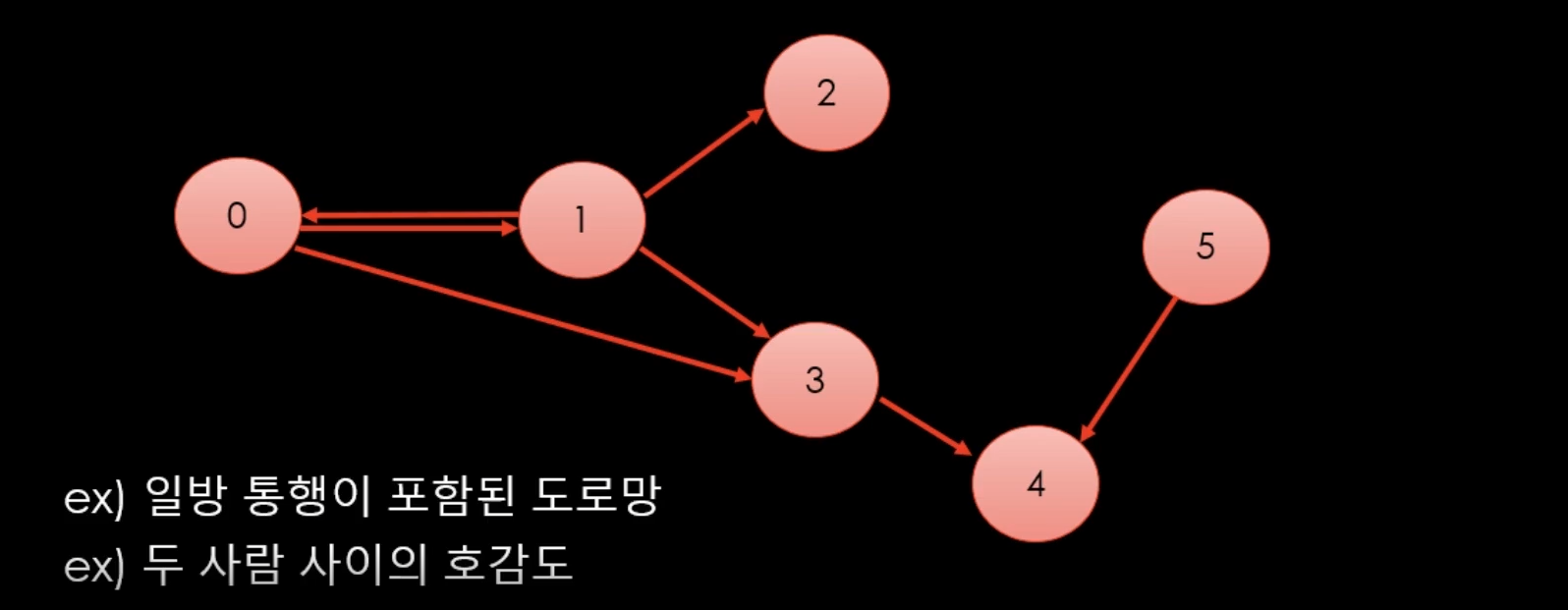
* 그래프 ex)



* 가중치 그래프

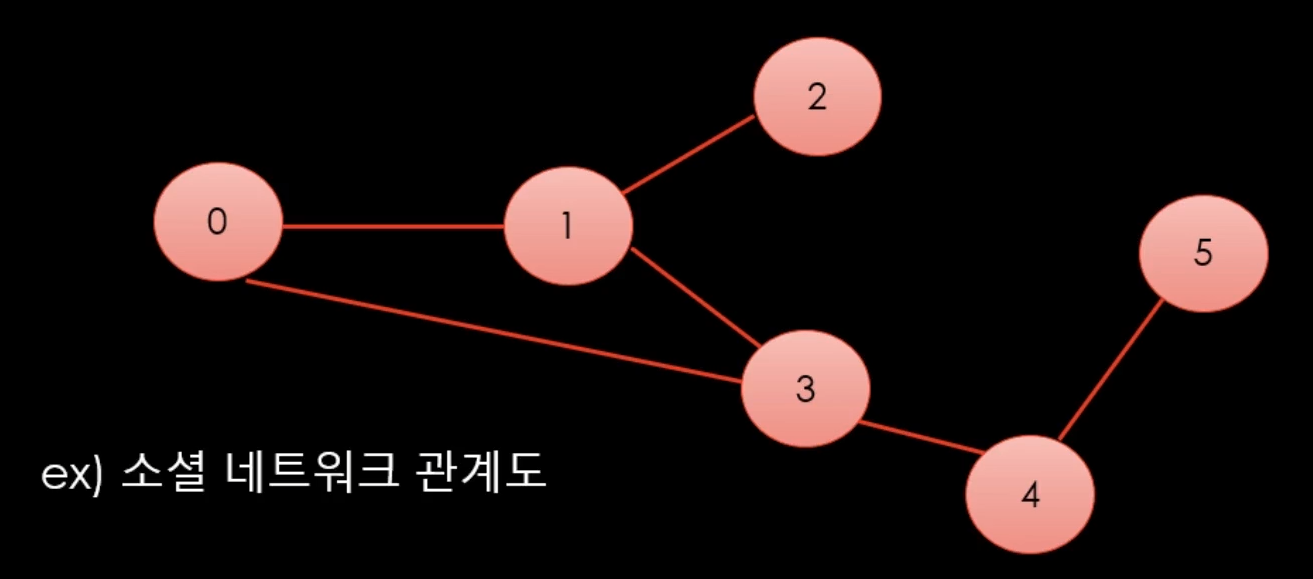


* 방향 그래프



* 코드로 어떻게 구현을 할까?

1. 탐색(DFS, BFS)



* DFS : 깊이 우선 탐색
* 위의 그래프로 봤을 때, 연결 된 곳으로 무조건 가고, 나중에 되돌아옴 (012345, 034512, 013452 등)
* 행렬과 리스트로 그래프 구현

화살이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

* + 행렬로 그래프를 만들었을 때 DFS 구현

텍스트이(가) 표시된 사진

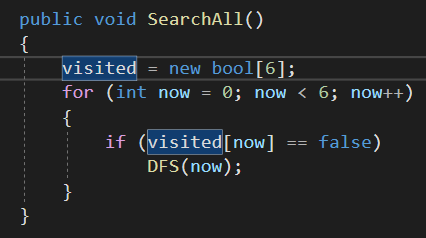
자동 생성된 설명

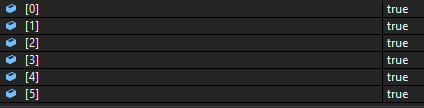
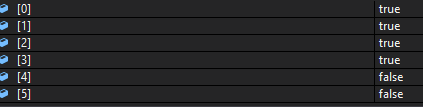
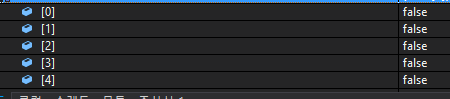
* + 리스트로 그래프를 만들었을 때 DFS 구현

텍스트이(가) 표시된 사진

자동 생성된 설명

* 근데 만약 그래프가 모두 연결되어 있지 않다면?





* BFS : 너비 우선 탐색 (최단 거리 탐색)
  + 행렬로 그래프를 만들었을 때, BFS 구현

텍스트이(가) 표시된 사진

자동 생성된 설명

* 추가적인 정보 부모가 누구인가? 거리가 얼만큼인가 등의 정보를 넣어 활용할 수 있다.

1. BFS를 사용한 길찾기 알고리즘