

ANSI 알고리즘 스터디

02

BFS

01 그래프 탐색 알고리즘

02 너비 우선 탐색 BFS

03 문제 풀어보기



01

그래프 탐색 알고리즘

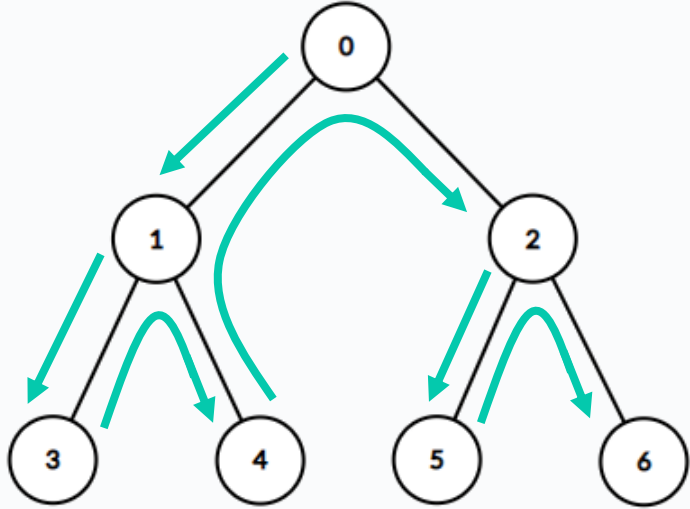
>>>>



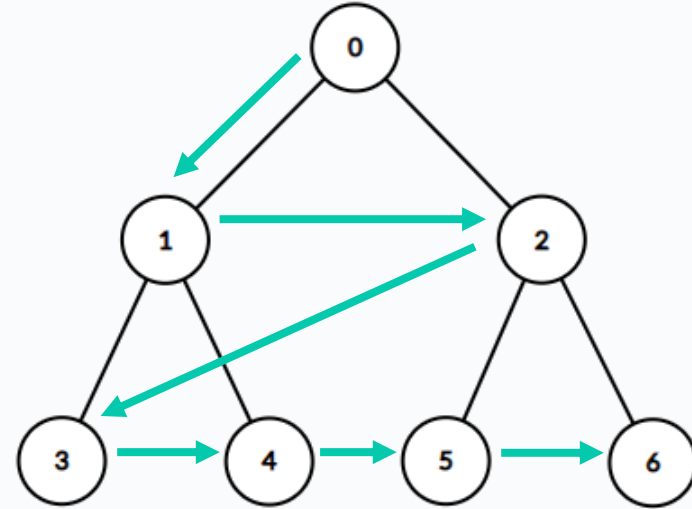
그래프 탐색 알고리즘

그래프의 모든 정점들을 특정한 순서에 따라 방문하는 알고리즘

- 깊이 우선 탐색 **DFS** (Depth First Search)



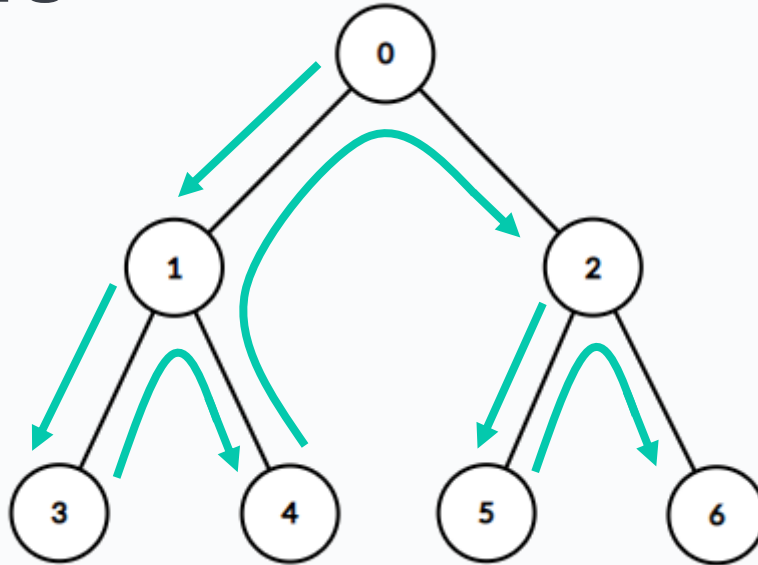
- 너비 우선 탐색 **BFS** (Breadth First Search)



깊이 우선 탐색 DFS

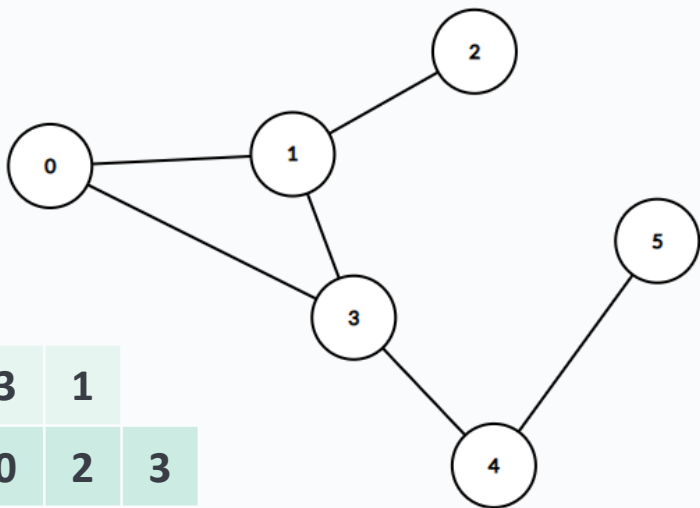
- 현재 정점에서 갈 수 있는 정점들까지 **들어가며** 탐색

✓ 한 방향으로 계속 갈 수 있을 때까지 탐색하다 막히면, 가장 가까운 갈림길로 돌아와서 다른 방향으로 탐색을 진행



깊이 우선 탐색 DFS

- DFS 구현해보기



graph[0]	3	1	
graph[1]	0	2	3
graph[2]	1		
graph[3]	0	4	1
graph[4]	3	5	
graph[5]	4		

```
1  vector<int> graph[6];
2  int visited[6];
3
4  void dfs(int cur)
5  {
6      visited[cur] = 1;
7
8      for (int i = 0; i < graph[cur].size(); i++)
9      {
10         int next = graph[cur][i];
11         if (visited[next] == 0)
12             dfs(next);
13     }
14 }
```



02

너비 우선 탐색 BFS

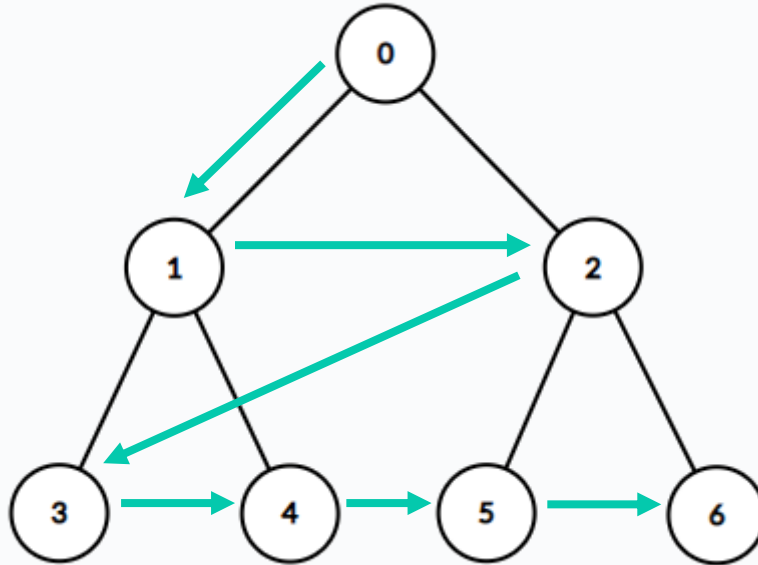
>>>>



너비 우선 탐색 BFS

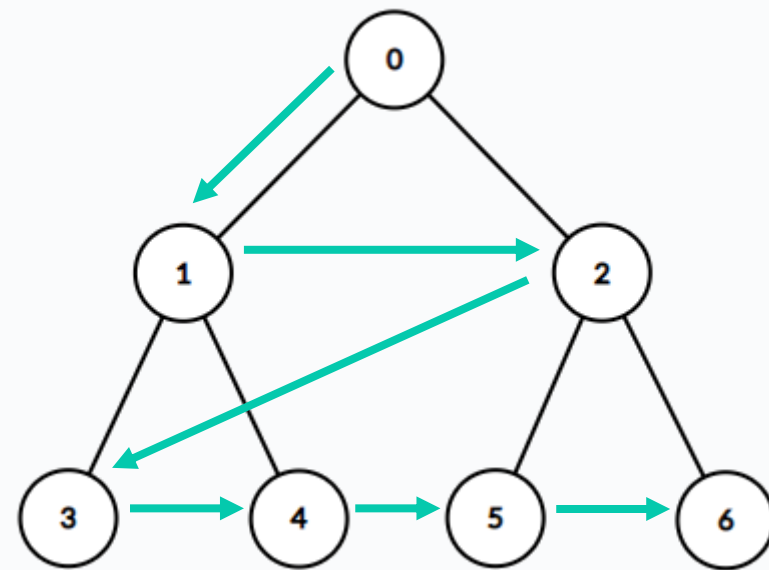
- 현재 정점에서 인접한 연결된 정점부터 탐색

✓ 시작 정점에 **가까운 정점**부터 넓게 탐색하여 먼 정점은 가장 나중에 방문



너비 우선 탐색 BFS

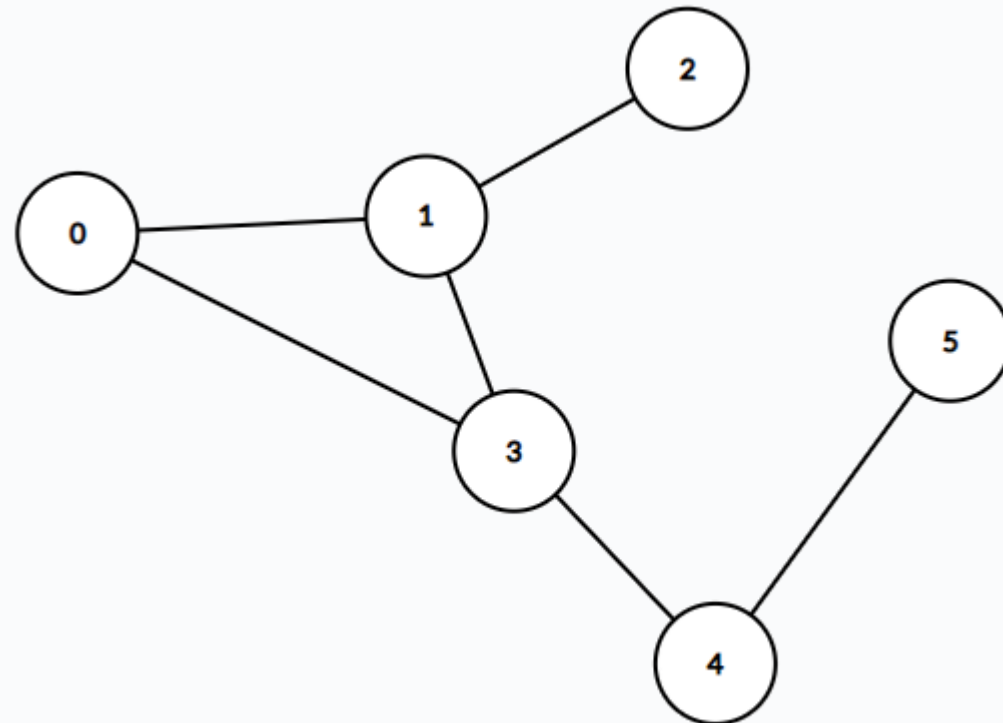
1. 큐에서 정점 하나를 뺀다.
2. 빼낸 정점과 인접한 정점들을 큐에 넣는다.



너비 우선 탐색 BFS

queue<int> q

탐색 순서

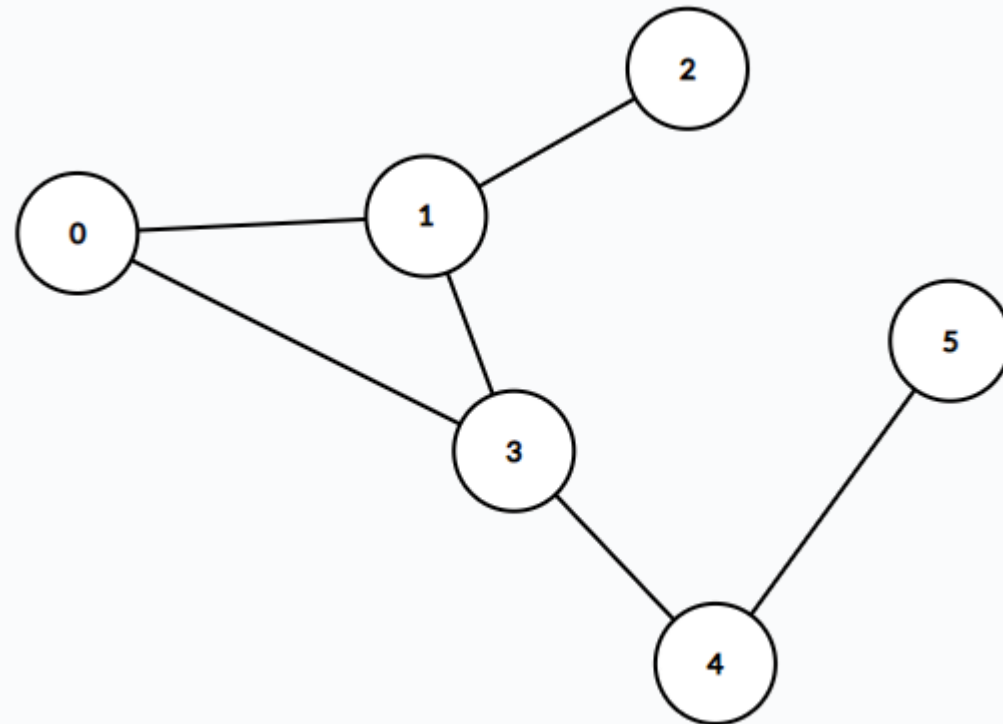


너비 우선 탐색 BFS

queue<int> q

0

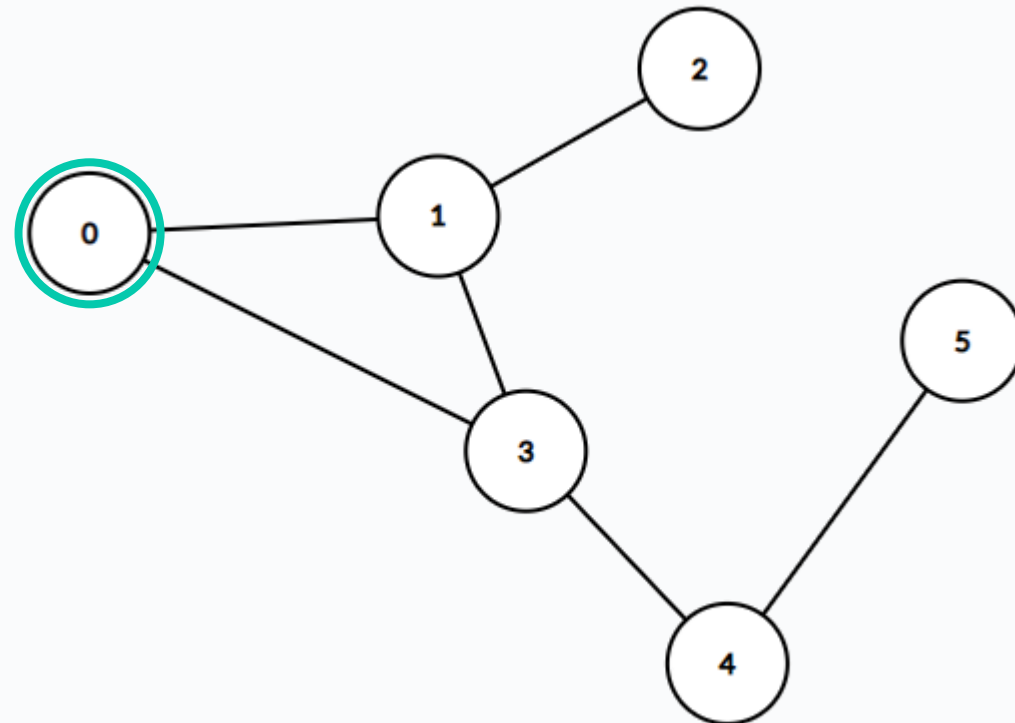
탐색 순서



너비 우선 탐색 BFS

queue<int> q

탐색 순서 0

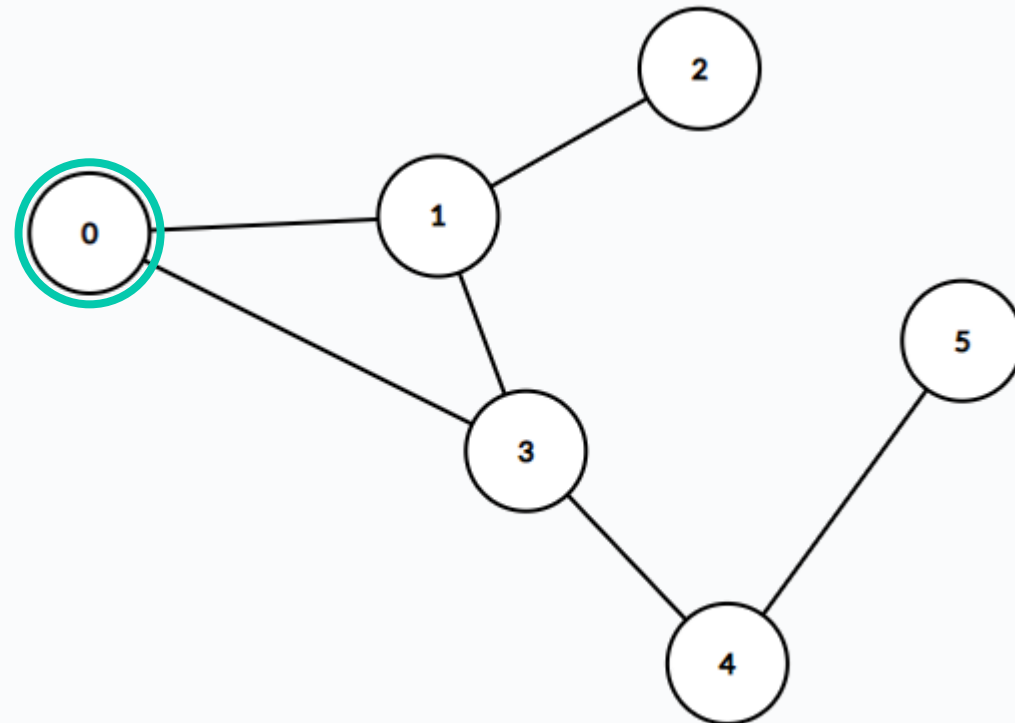


너비 우선 탐색 BFS

queue<int> q

1 3

방문 순서 0

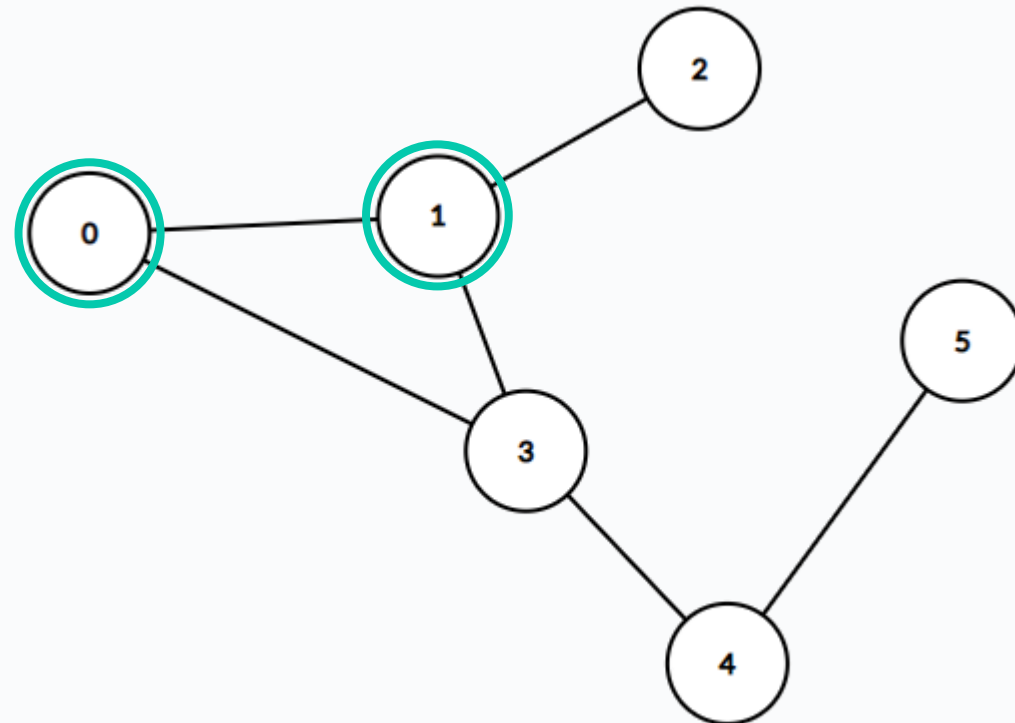


너비 우선 탐색 BFS

queue<int> q



방문 순서 0 -> 1

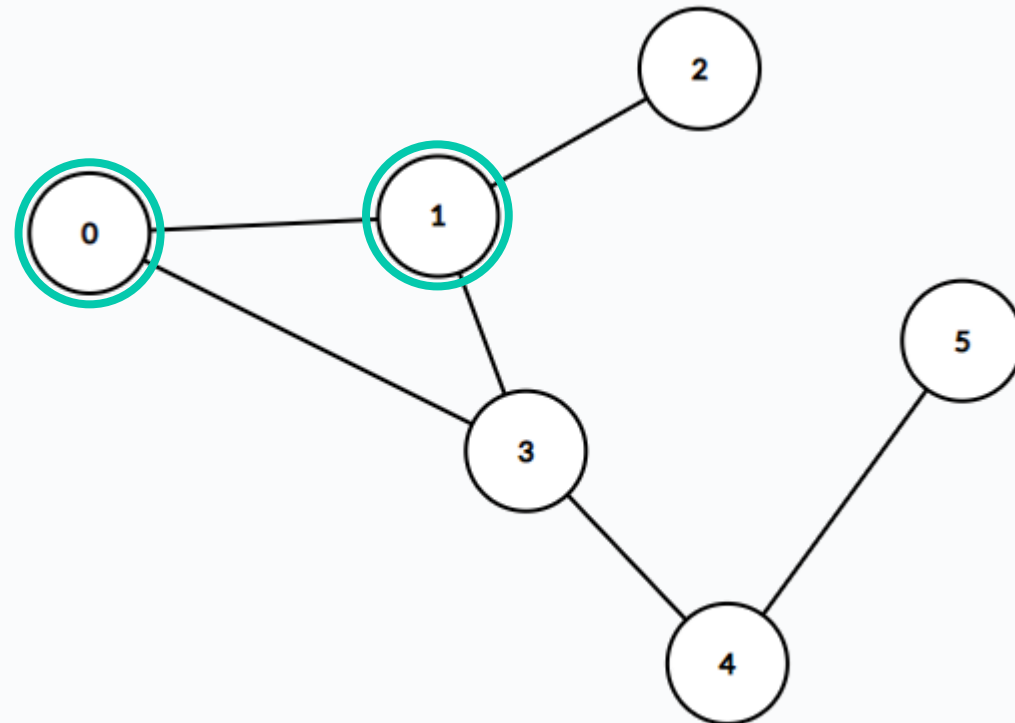


너비 우선 탐색 BFS

queue<int> q

3 2

방문 순서 0 -> 1

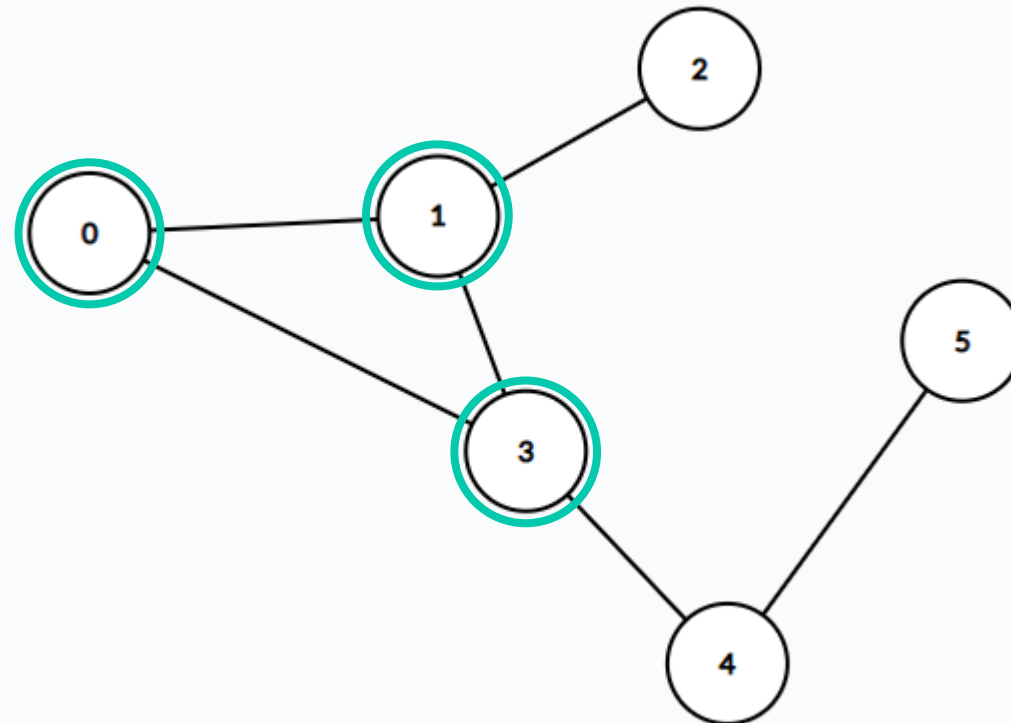


너비 우선 탐색 BFS

queue<int> q



방문 순서 0 -> 1 -> 3

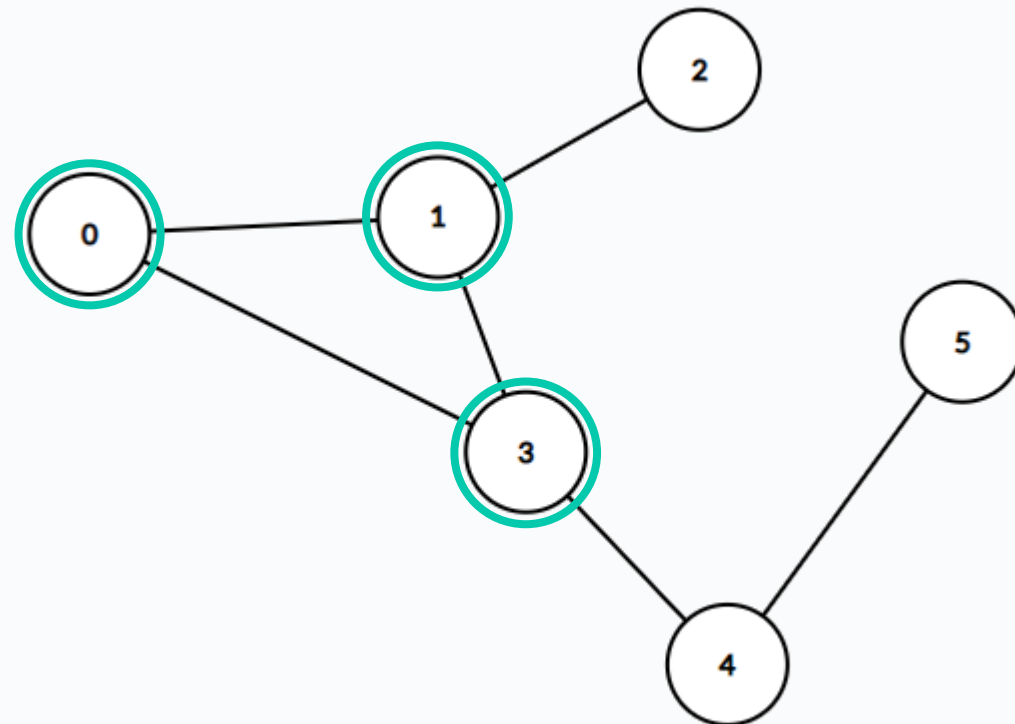


너비 우선 탐색 BFS

queue<int> q

2 4

방문 순서 0 -> 1 -> 3

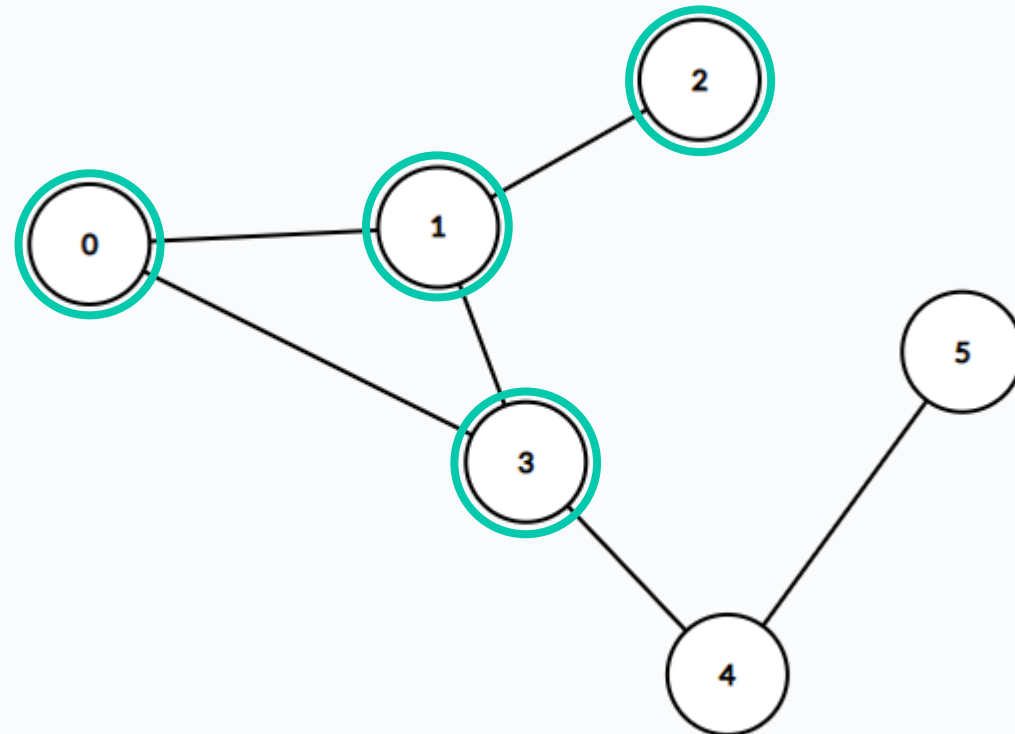


너비 우선 탐색 BFS

queue<int> q



방문 순서 0 -> 1 -> 3 -> 2

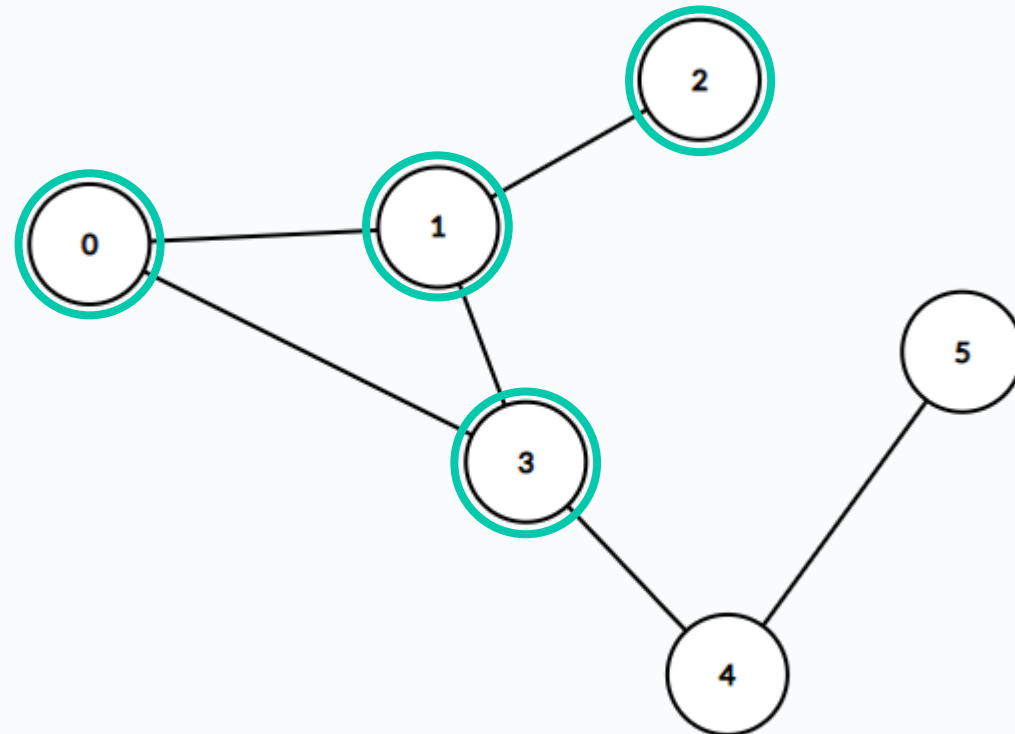


너비 우선 탐색 BFS

queue<int> q

4

방문 순서 0 -> 1 -> 3 -> 2

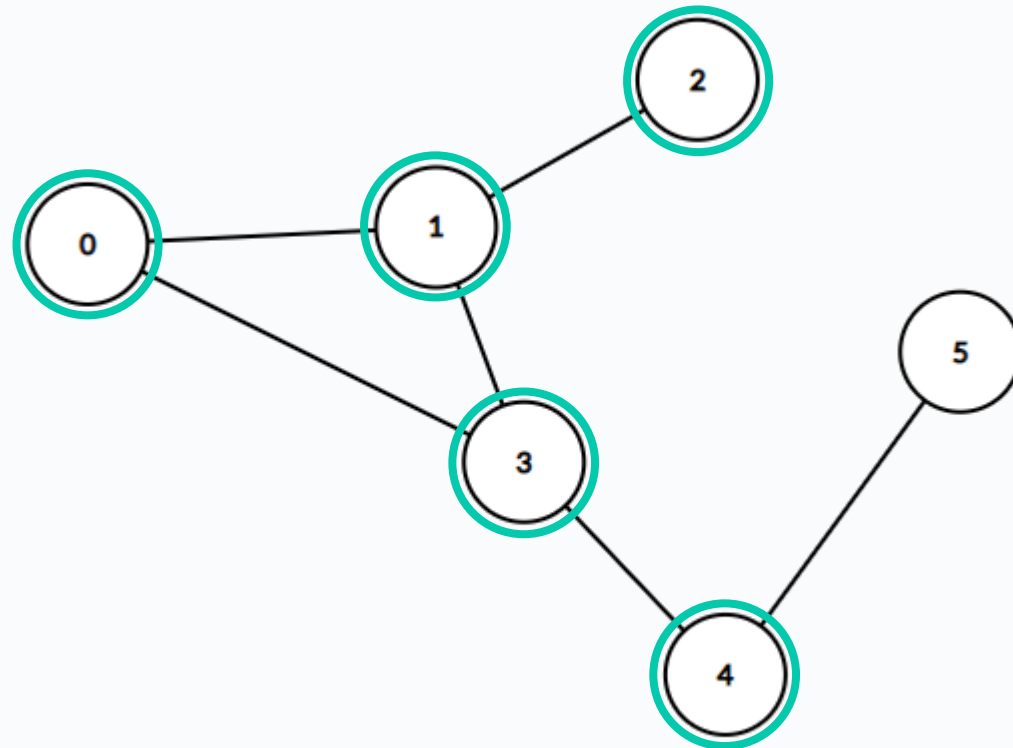


너비 우선 탐색 BFS

queue<int> q



방문 순서 0 -> 1 -> 3 -> 2 -> 4

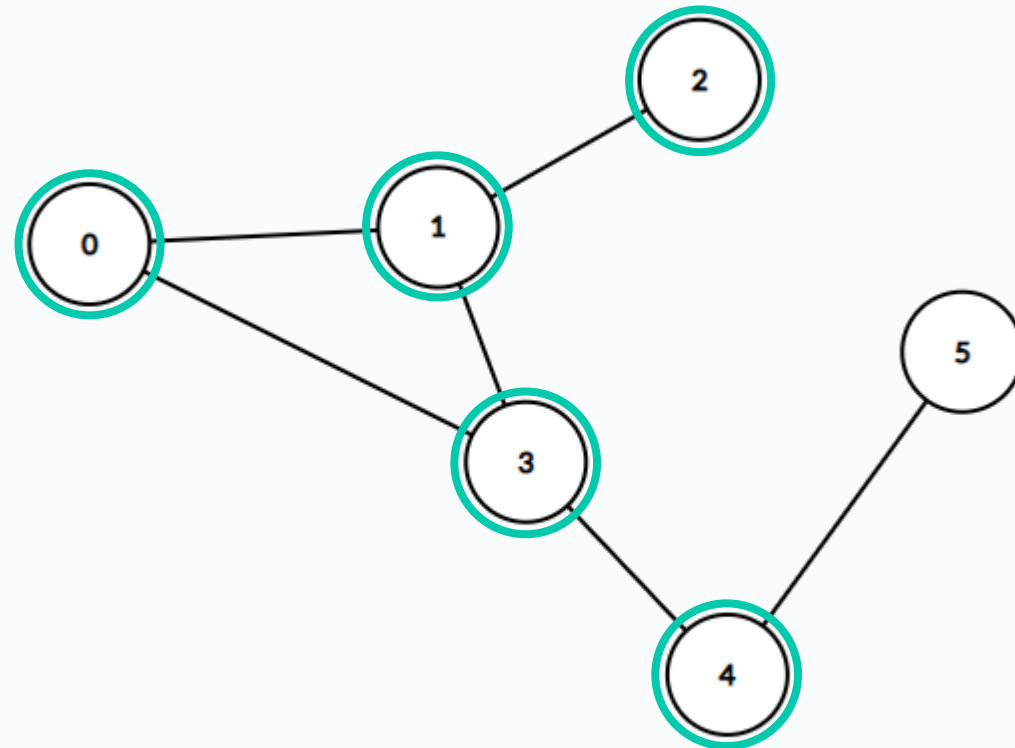


너비 우선 탐색 BFS

queue<int> q

5

방문 순서 0 -> 1 -> 3 -> 2 -> 4



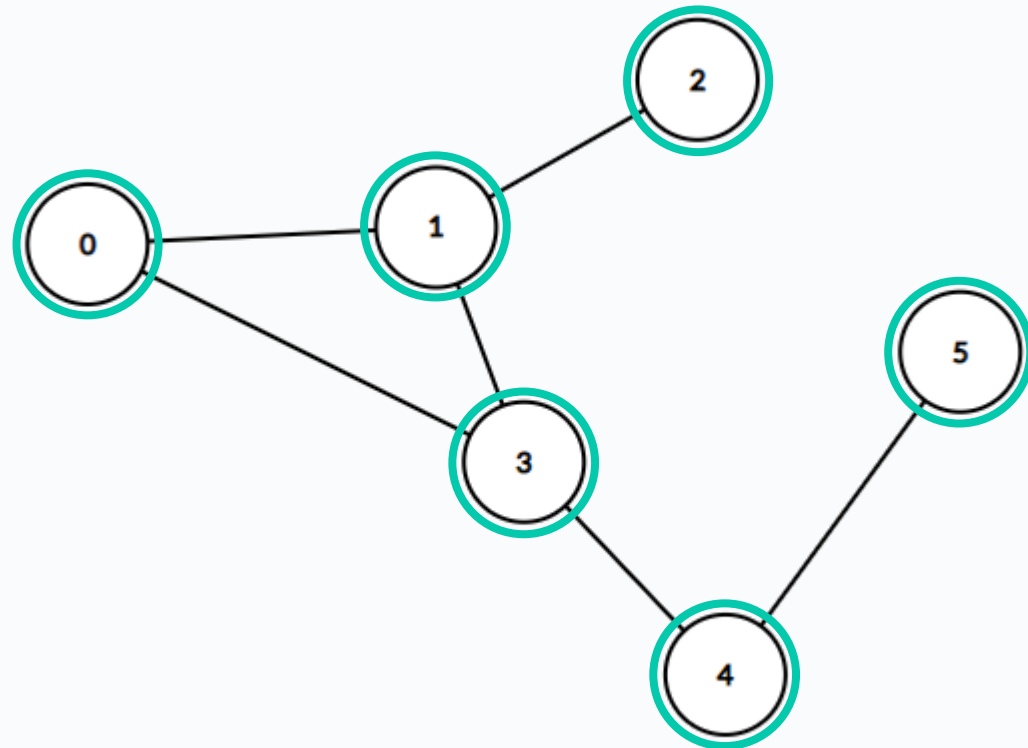
너비 우선 탐색 BFS

queue<int> q



5

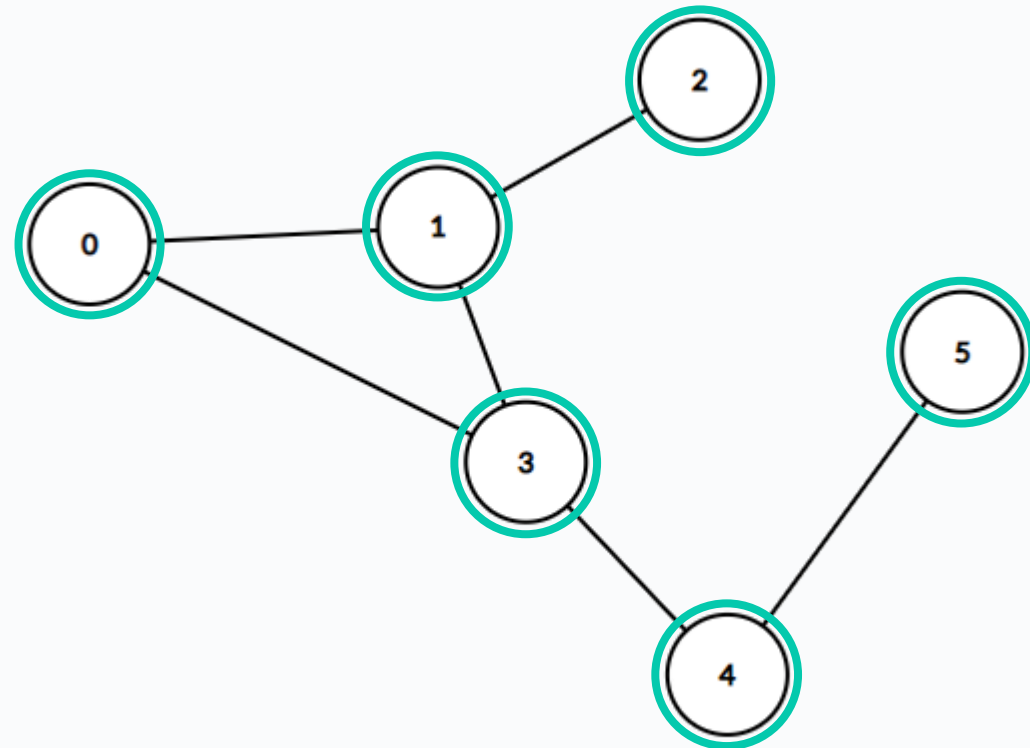
방문 순서 0 -> 1 -> 3 -> 2 -> 4 -> 5



너비 우선 탐색 BFS

queue<int> q

방문 순서 0 -> 1 -> 3 -> 2 -> 4 -> 5



너비 우선 탐색 BFS

- BFS 구현해보기

`vector<int> graph[6];`

`queue<int> q;`

`int visited[6];`

```
1 void bfs(int start) {
2     q.push(start);
3     visited[start] = 1;
4     while(!q.empty()) {
5         int cur = q.front();
6         q.pop();
7         for(int i = 0; i < graph[cur].size(); i++) {
8             int next = graph[cur][i];
9             if(visited[next] == 0) {
10                 q.push(next);
11                 visited[next] = 1;
12             }
13         }
14     }
15 }
```



03

문제 풀어보기



문제 풀어보기

- #1260 DFS와 BFS (<https://www.acmicpc.net/problem/1260>)
- #2178 미로 탐색 (<https://www.acmicpc.net/problem/2178>)
- #2644 촌수계산 (<https://www.acmicpc.net/problem/2644>)
- #7576 토마토 (<https://www.acmicpc.net/problem/7576>)
- #7562 나이트의 이동 (<https://www.acmicpc.net/problem/7562>)



+

>>>>



수고하셨습니다

다음 강의는 BFS입니다



>



+



ANSi