

컴퓨터공학 All in One

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)
제 34강 - 깊이 우선 탐색

학습 목표

깊이 우선 탐색

- 1) 깊이 우선 탐색의 개념을 이해합니다.
- 2) 깊이 우선 탐색을 C언어를 이용하여 구현할 수 있습니다.

깊이 우선 탐색

깊이 우선 탐색

깊이 우선 탐색(Depth First Search)은 탐색을 함에 있어서 보다 깊은 것을 우선적으로 하여 탐색하는 알고리즘입니다. 깊이 우선 탐색은 기본적으로 전체 노드를 맹목적으로 탐색하고자 할 때 사용합니다. 깊이 우선 탐색 알고리즘은 스택(Stack) 자료구조에 기초합니다.

깊이 우선 탐색은 빠르게 모든 경우의 수를 탐색하고자 할 때 쉽게 사용할 수 있습니다.

깊이 우선 탐색

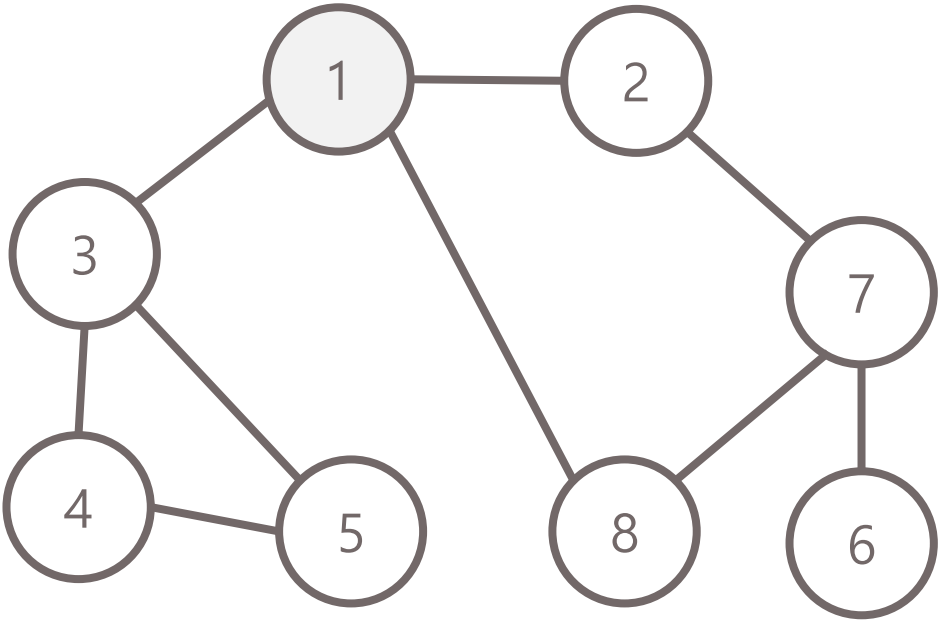
깊이 우선 탐색

1. 탐색 시작 노드를 스택에 삽입하고 방문 처리를 합니다.
2. 스택의 최상단 노드에게 방문하지 않은 인접 노드가 하나라도 있으면 그 노드를 스택에 넣고 방문 처리를 합니다. 방문하지 않은 인접 노드가 없으면 스택에서 최상단 노드를 꺼냅니다.
3. 2번의 과정을 더 이상 수행할 수 없을 때까지 반복합니다.

깊이 우선 탐색

깊이 우선 탐색

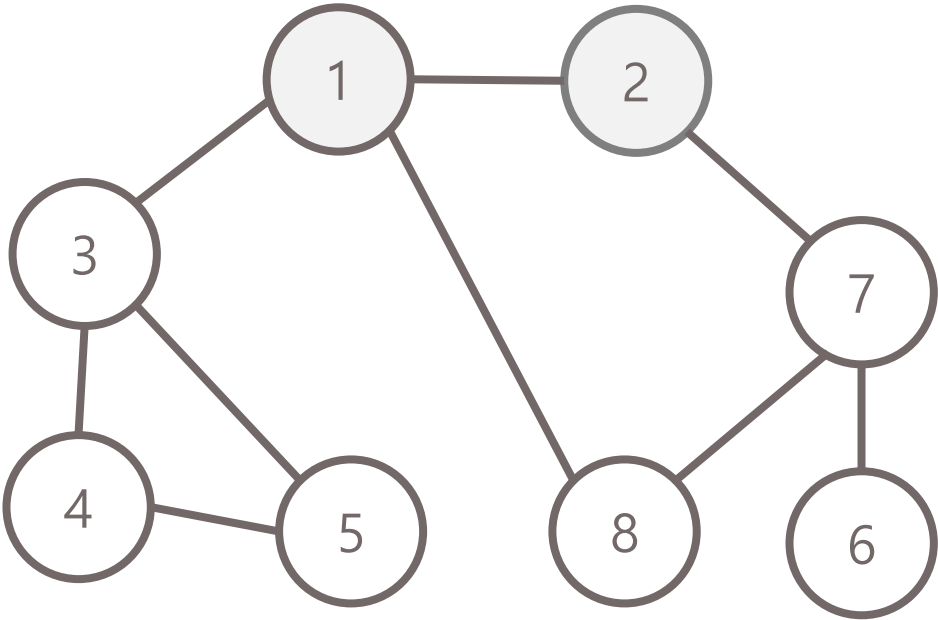
1



깊이 우선 탐색

깊이 우선 탐색

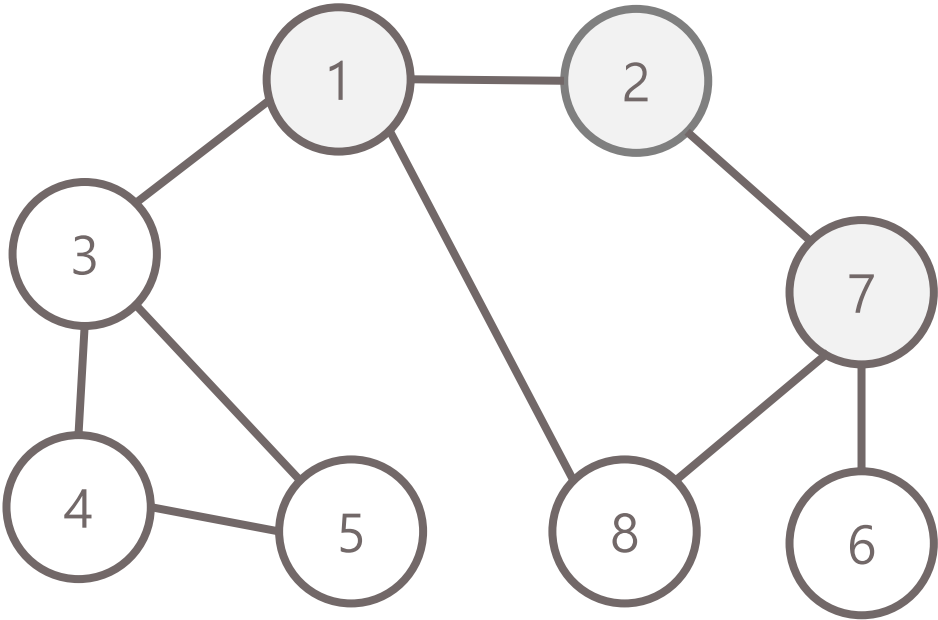
2
1



깊이 우선 탐색

깊이 우선 탐색

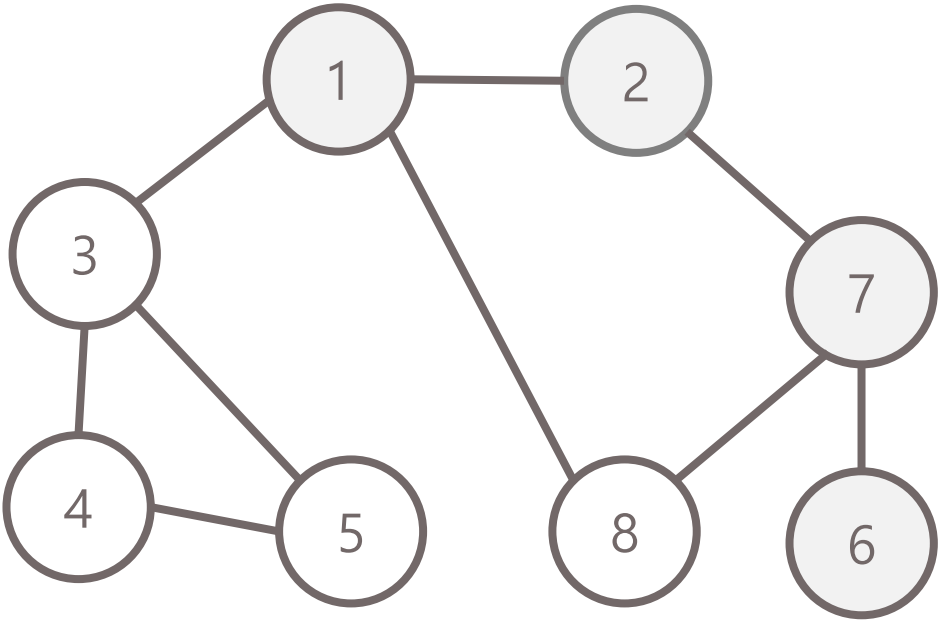
7
2
1



깊이 우선 탐색

깊이 우선 탐색

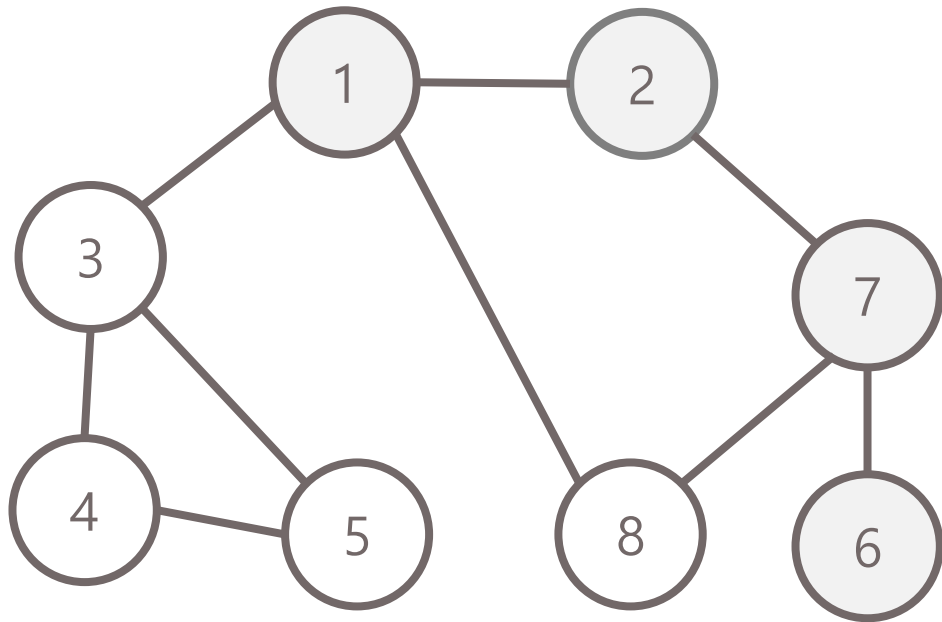
6
7
2
1



깊이 우선 탐색

깊이 우선 탐색

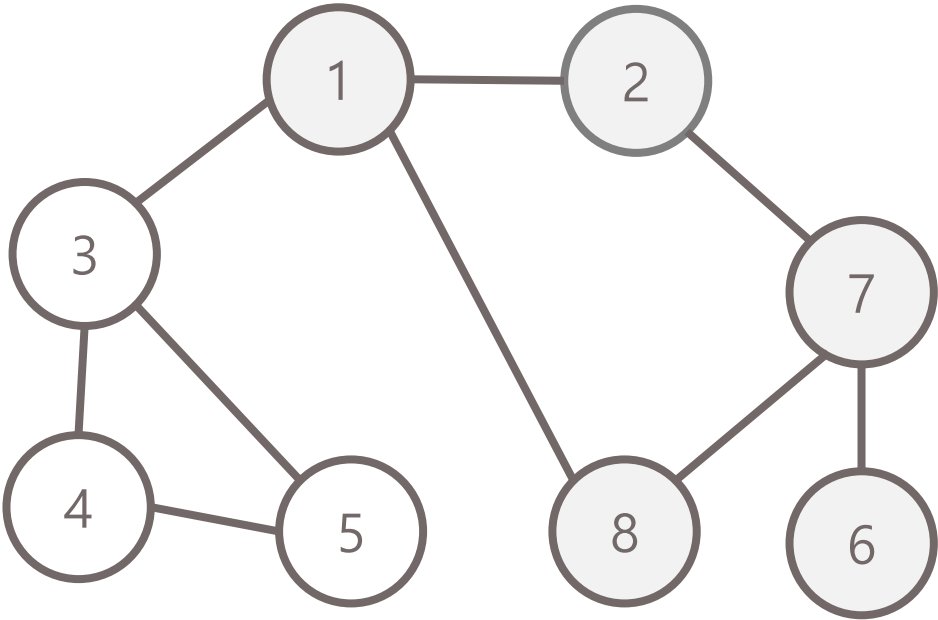
7
2
1



깊이 우선 탐색

깊이 우선 탐색

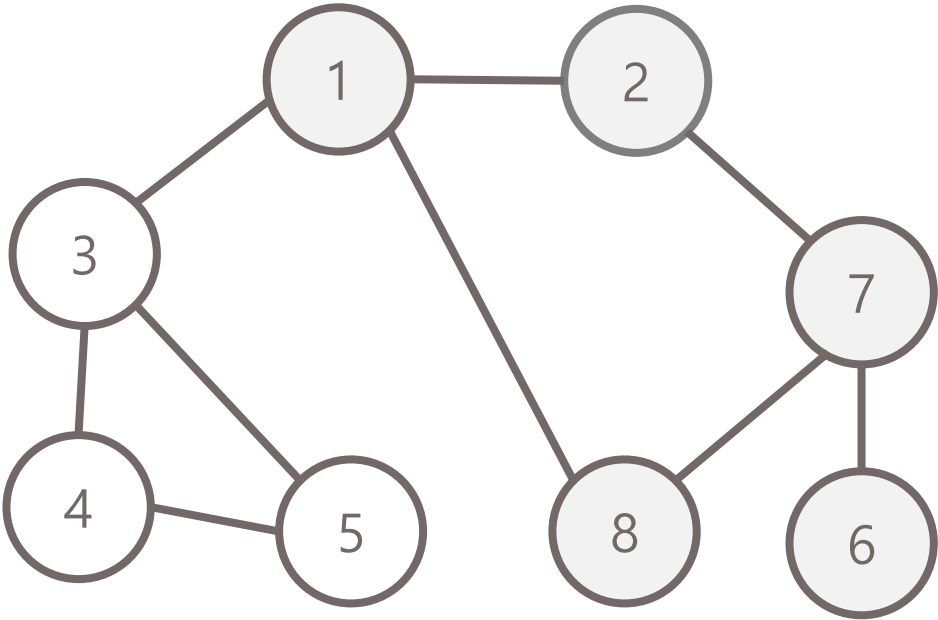
8
7
2
1



깊이 우선 탐색

깊이 우선 탐색

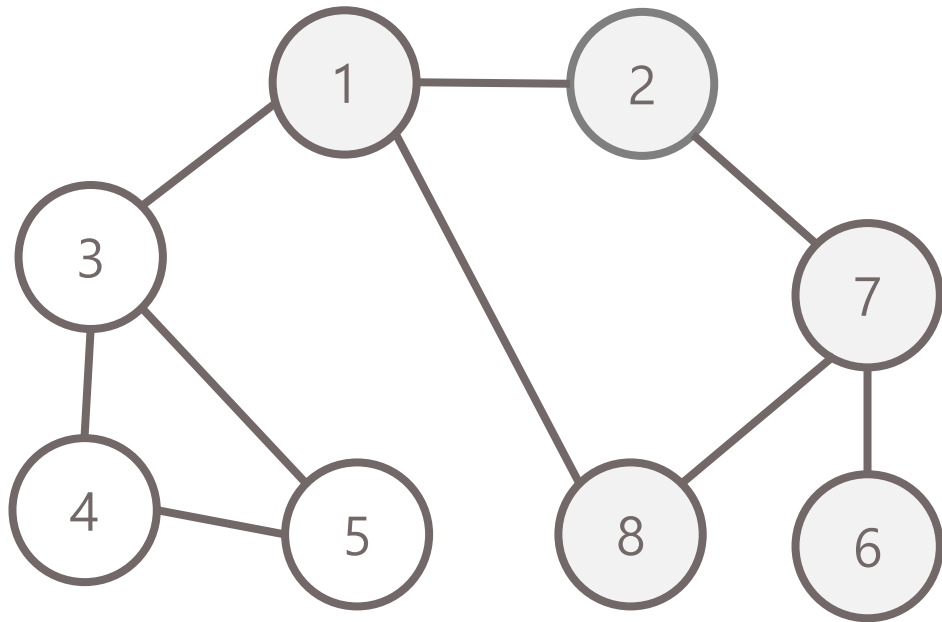
7
2
1



깊이 우선 탐색

깊이 우선 탐색

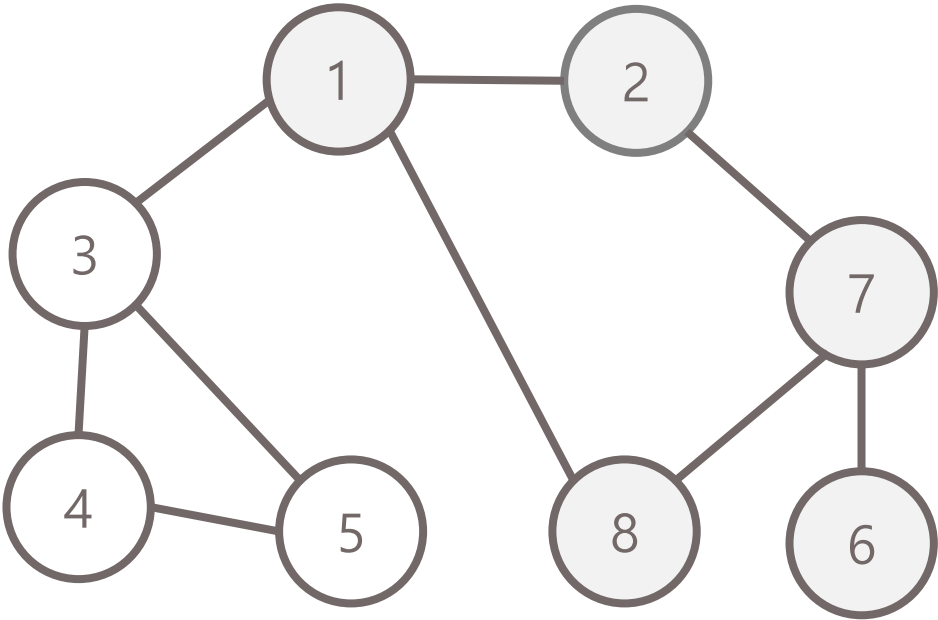
2
1



깊이 우선 탐색

깊이 우선 탐색

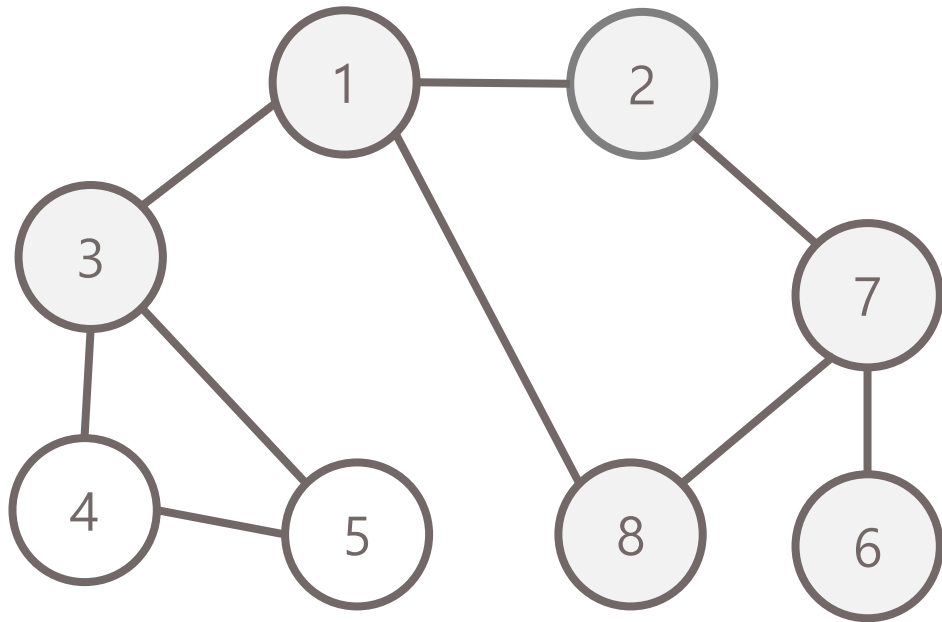
1



깊이 우선 탐색

깊이 우선 탐색

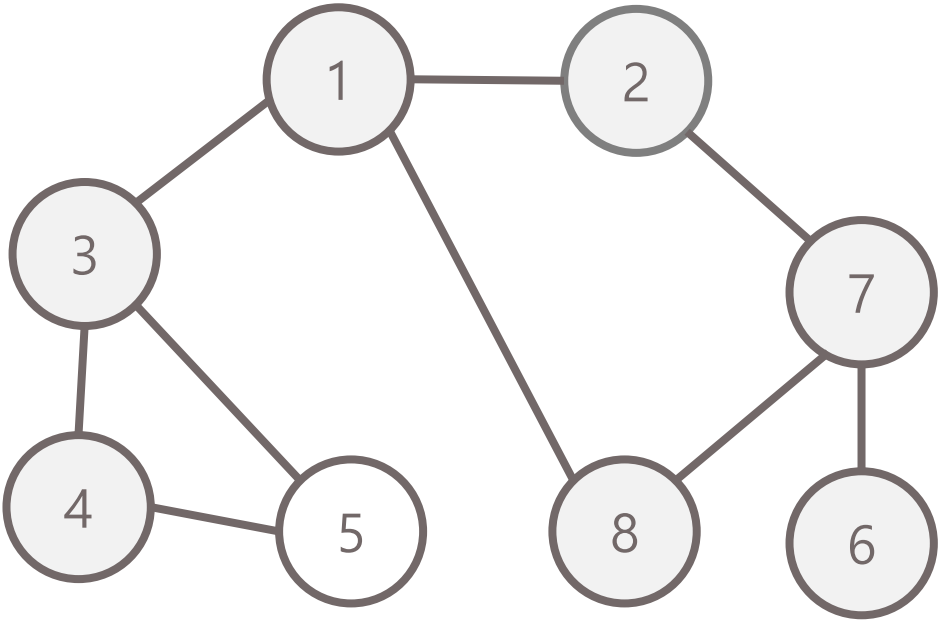
3
1



깊이 우선 탐색

깊이 우선 탐색

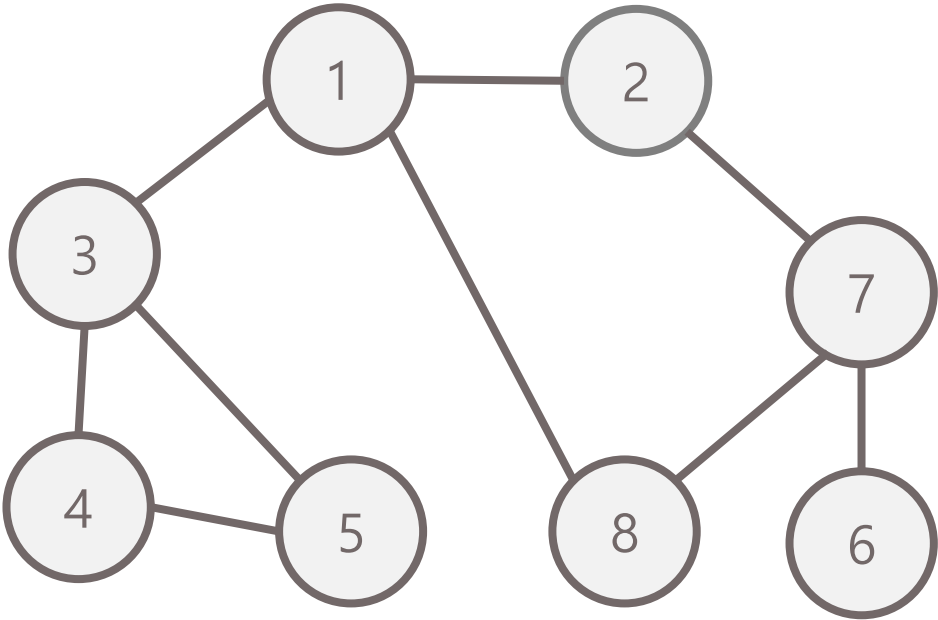
4
3
1



깊이 우선 탐색

깊이 우선 탐색

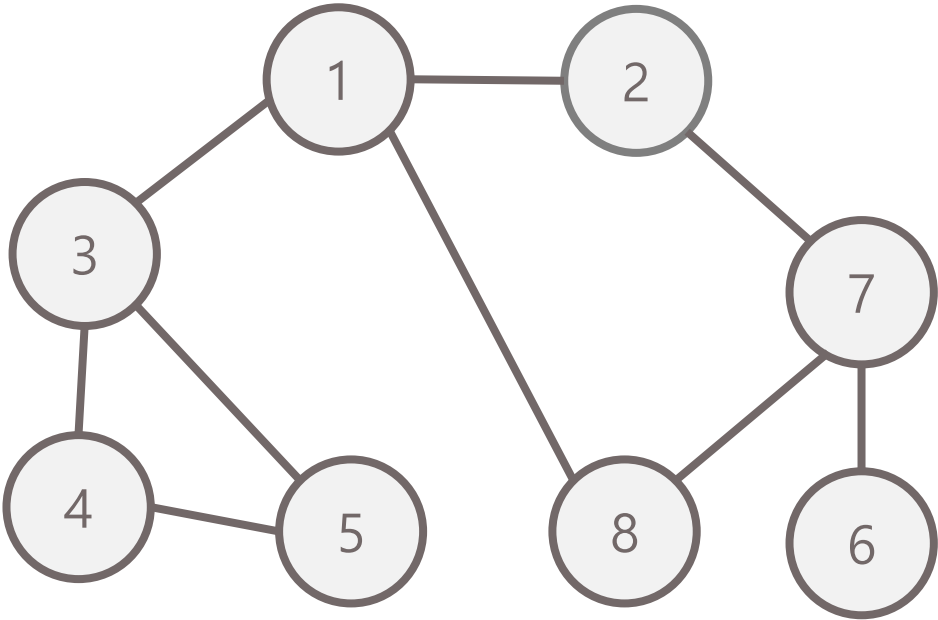
5
4
3
1



깊이 우선 탐색

깊이 우선 탐색

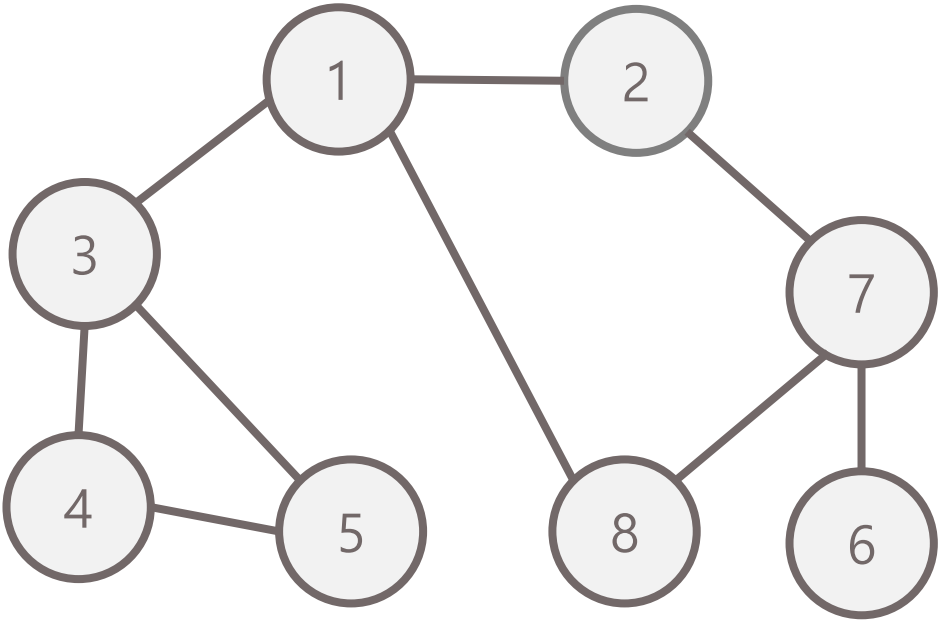
4
3
1



깊이 우선 탐색

깊이 우선 탐색

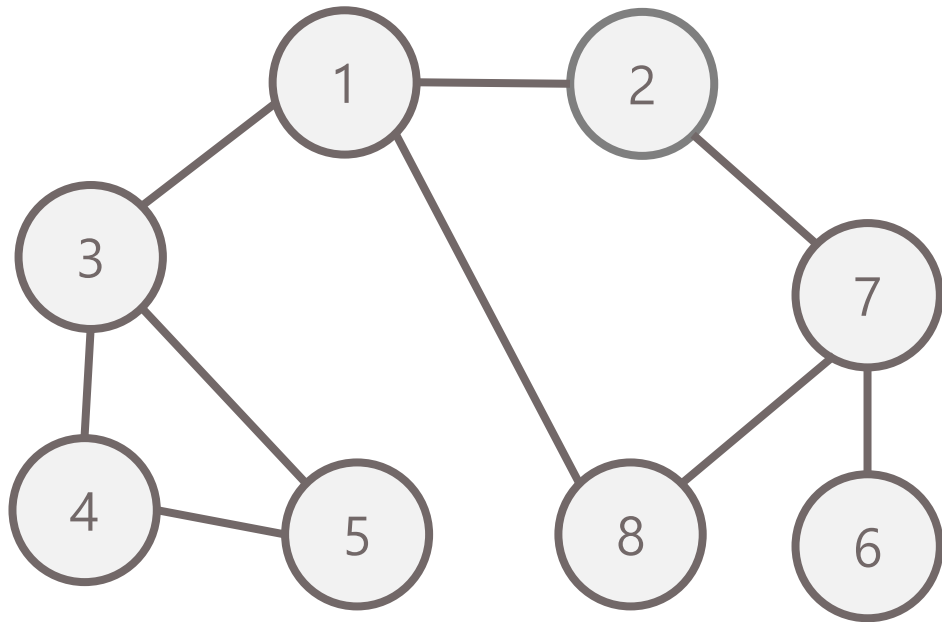
3
1



깊이 우선 탐색

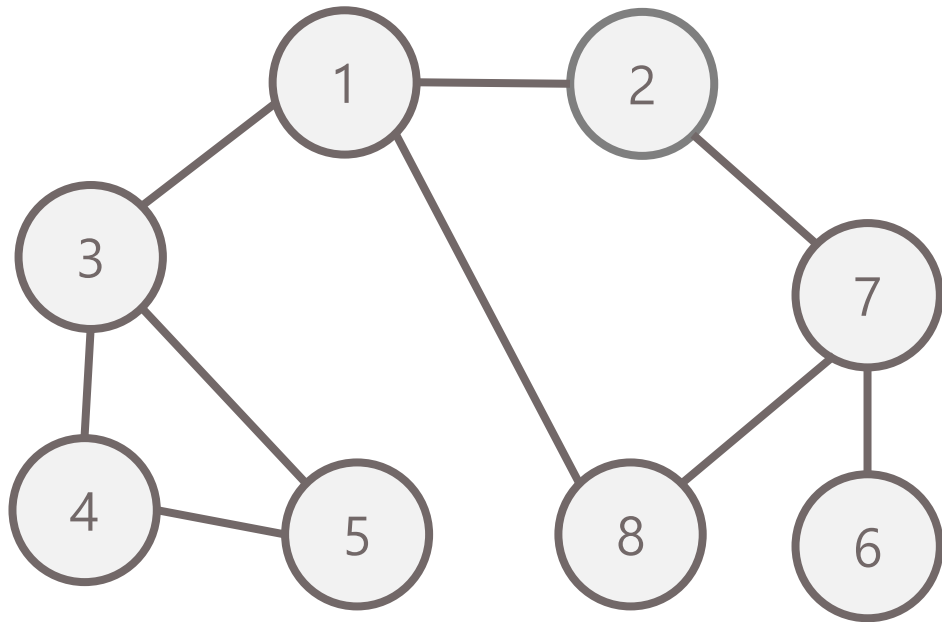
깊이 우선 탐색

1



깊이 우선 탐색

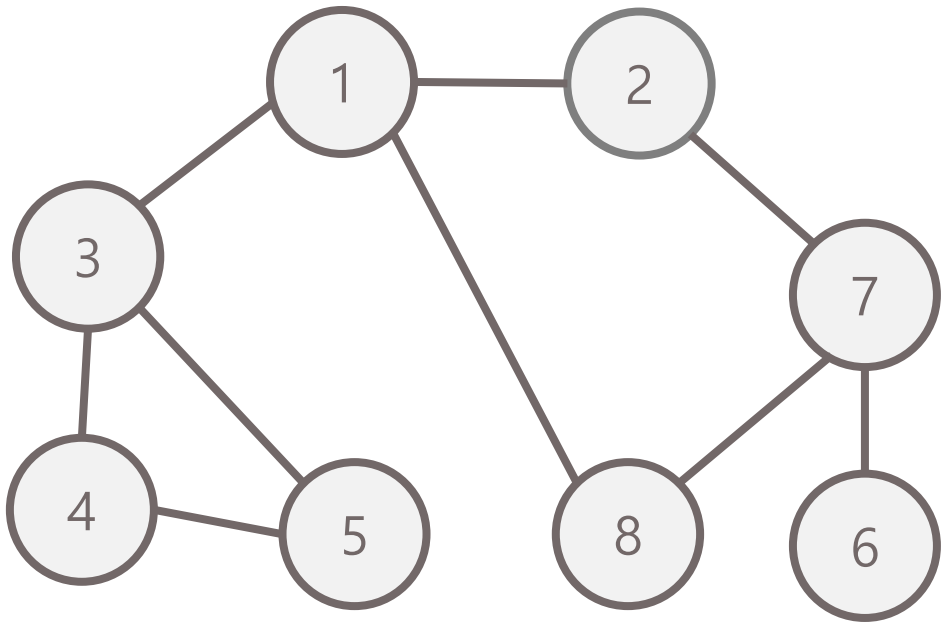
깊이 우선 탐색



깊이 우선 탐색

깊이 우선 탐색

방문 순서: $1 \rightarrow 2 \rightarrow 7 \rightarrow 6 \rightarrow 8 \rightarrow 3 \rightarrow 4 \rightarrow 5$



깊이 우선 탐색 알고리즘은 스택(Stack) 자료구조에 기초한다는 점에서 구현이 간단합니다. 실제로는 스택을 쓰지 않아도 되며 탐색을 수행함에 있어서 $O(N)$ 의 시간이 소요됩니다.

깊이 우선 탐색

깊이 우선 탐색 1) 연결 리스트 정의

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#define MAX_SIZE 1001

typedef struct {
    int index;
    struct Node *next;
} Node;

Node** a;
int n, m, c[MAX_SIZE];
```

깊이 우선 탐색

깊이 우선 탐색 2) 연결 리스트 삽입 함수

```
void addFront(Node *root, int index) {  
    Node *node = (Node*)malloc(sizeof(Node));  
    node->index = index;  
    node->next = root->next;  
    root->next = node;  
}
```

깊이 우선 탐색

깊이 우선 탐색 3) 깊이 우선 탐색 함수

```
void dfs(int x) {  
    if (c[x]) return;  
    c[x] = 1;  
    printf("%d ", x);  
    Node *cur = a[x]->next;  
    while (cur != NULL) {  
        int next = cur->index;  
        dfs(next);  
        cur = cur->next;  
    }  
}
```


깊이 우선 탐색

깊이 우선 탐색 4) 깊이 우선 탐색 이용해보기

```
int main(void) {
    scanf("%d %d", &n, &m);
    a = (Node**)malloc(sizeof(Node*) * (MAX_SIZE));
    for (int i = 1; i <= n; i++) {
        a[i] = (Node*)malloc(sizeof(Node));
        a[i]->next = NULL;
    }
    for (int i = 0; i < m; i++) {
        int x, y;
        scanf("%d %d", &x, &y);
        addFront(a[x], y);
        addFront(a[y], x);
    }
    dfs(1);
    system("pause");
    return 0;
}
```

깊이 우선 탐색

깊이 우선 탐색

1) 깊이 우선 탐색은 $O(N)$ 의 시간이 소요되는 전수 탐색 알고리즘입니다.