

201201871 서현택

### 1. 목표

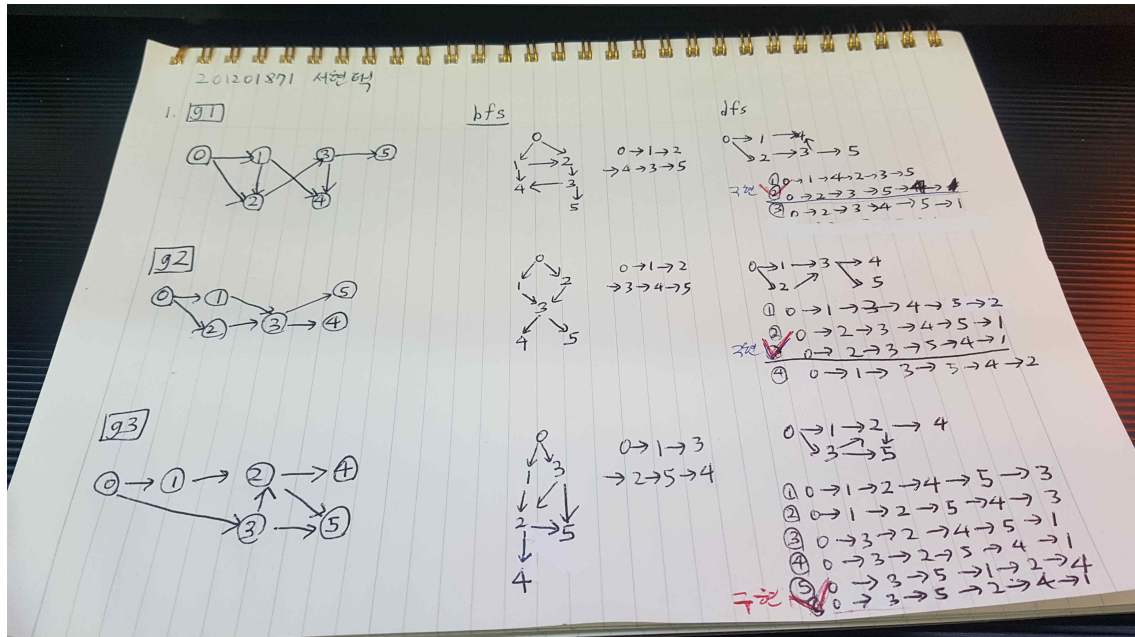
최단경로 알고리즘 구현(BFS, DFS)

### 2. 과제를 해결하는 방법

- (1) 실습 내용 기반으로 BFS, DFS 알고리즘 구현
- (2) 그림을 그려보면서 알고리즘 이해

### 3. 과제를 해결한 방법

- (1) 그림을 통한 이해



### (2) BFS

list형 변수 queue와 set형 변수 visited를 생성한 뒤, start\_node를 추가한다. 그 후, start\_node를 출력한다. queue의 길이가 0보다 클 때 동안, visiting에 queue의 첫 번째 자료를 pop시킨 것을 저장한다. neighbor가 graph의 리스트에 있는지 확인하고, visited에 neighbor가 없다면 visited에 neighbor를 추가하고, neighbor를 출력한 뒤, queue에 neighbor를 저장한다.

### (3) DFS

list형 변수 stack과 set형 변수 visited를 생성한 뒤, stack에 start\_node를 삽입한다. stack의 길이가 0보다 클 때 동안, visiting에 stack을 pop한 값을 저장하고, visited에 visiting이 없다면 visited에 visiting을 추가한다. 그 후 visiting을 출력한 뒤, graph의 vert\_list에 포함되어있는 값을 stack에 저장한다.

#### (4) Graph 클래스

##### 1) init

Graph 클래스를 초기화한다.

##### 2) add\_vertex

Graph 클래스에 key를 통해 vertex를 추가한다.

##### 3) get\_vertex

Graph 클래스에 저장된 vertex의 list에서 해당되는 vertex를 불러온다.

##### 4) contains

vertex에 해당 값이 저장되어 있나 확인한다.

##### 5) add\_edge

vert\_list[f]에서 vert\_list[t]로의 edge를 생성한다.

##### 6) get\_vertices

vert\_list의 keys를 불러온다.

##### 7) iter

vert\_list.values의 iter 함수 값을 반환한다.

#### 4.결과화면

##### (1) 제공된 코드

```
[ G diet_Bfs ]
Visit 0
Visit 1
Visit 4
Visit 5
Visit 2
Visit 6
Visit 3
[ G diet_Dfs ]
Visit 0
Visit 5
Visit 2
Visit 3
Visit 4
Visit 1
Visit 6
```

##### (2) G1

```
[ G1 diet_Bfs ]  
Visit 0  
Visit 1  
Visit 2  
Visit 4  
Visit 3  
Visit 5  
[ G1 diet_Dfs ]  
Visit 0  
Visit 2  
Visit 3  
Visit 5  
Visit 4  
Visit 1
```

(3) G2

```
[ G2 diet_Bfs ]  
Visit 0  
Visit 1  
Visit 2  
Visit 3  
Visit 4  
Visit 5  
[ G2 diet_Dfs ]  
Visit 0  
Visit 2  
Visit 3  
Visit 5  
Visit 4  
Visit 1
```

(4) G3

```
[ 63 diet_Bfs ]
```

```
Visit 0
```

```
Visit 1
```

```
Visit 3
```

```
Visit 2
```

```
Visit 5
```

```
Visit 4
```

```
[ 63 diet_Dfs ]
```

```
Visit 0
```

```
Visit 3
```

```
Visit 5
```

```
Visit 2
```

```
Visit 4
```

```
Visit 1
```