```
std::cout<<“Hello, graph theory!”;
```

알고리즘의 꽃 그래프를 배워봅시다

그 전에 복습 겸 손 풀기 문제 두개만 풀어봅시다

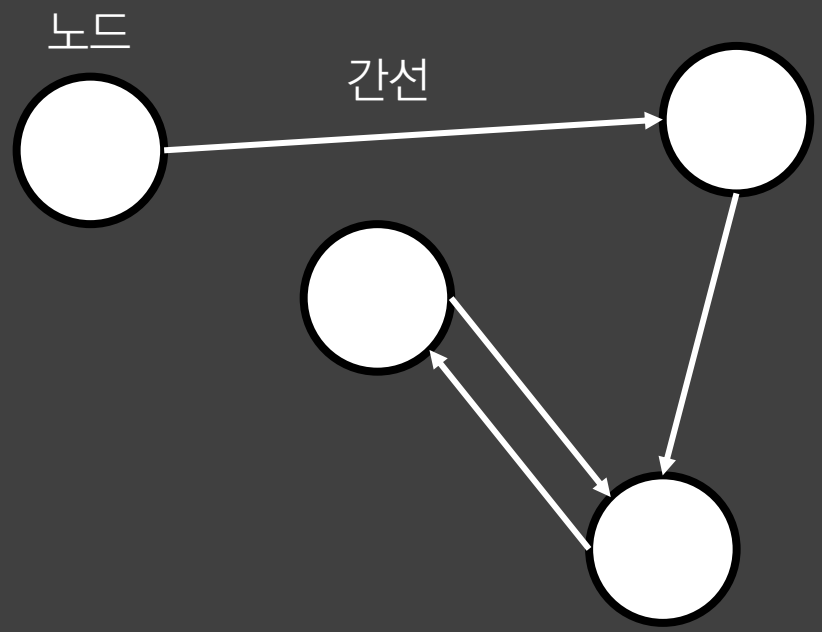
10845 – 큐 (Silver IV)

2748 – 피보나치 수 (Bronze I)

이번에 여러분들이 활용할 기법들이다

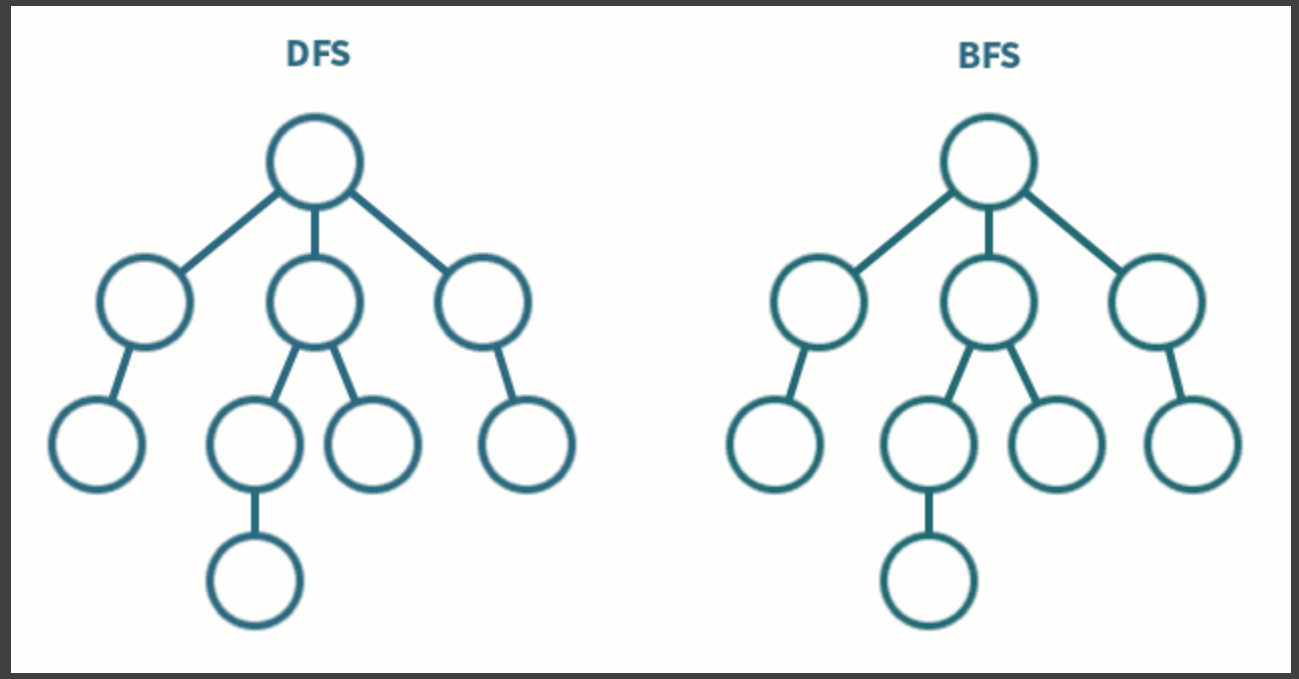
queue, 재귀함수, vector etc…

그래프의 기본 구조



양방향 그래프, 단방향 그래프

그래프 탐색 기법 – DFS, BFS



DFS – 깊이 우선 탐색 (Depth First Search)

그림판 ㄱㄱ

직접 구현해보면서 익혀보자

1260 - DFS와 BFS (Silver II)

BFS – 너비 우선 탐색 (Breadth First Search)

그림판 ㄱㄱ

마저 구현해보면서 익혀보자

1260 - DFS와 BFS (Silver II)

이번엔 여러분들이 풀어보자

2606 – 바이러스 (Silver III)

이번엔 이중 배열이다

1012 – 유기농 배추 (Silver II)

이중 배열의 인접 노드 탐색은 어떻게 해야할까?

2차원 배열에서의 인접 노드 : 상,하,좌,우

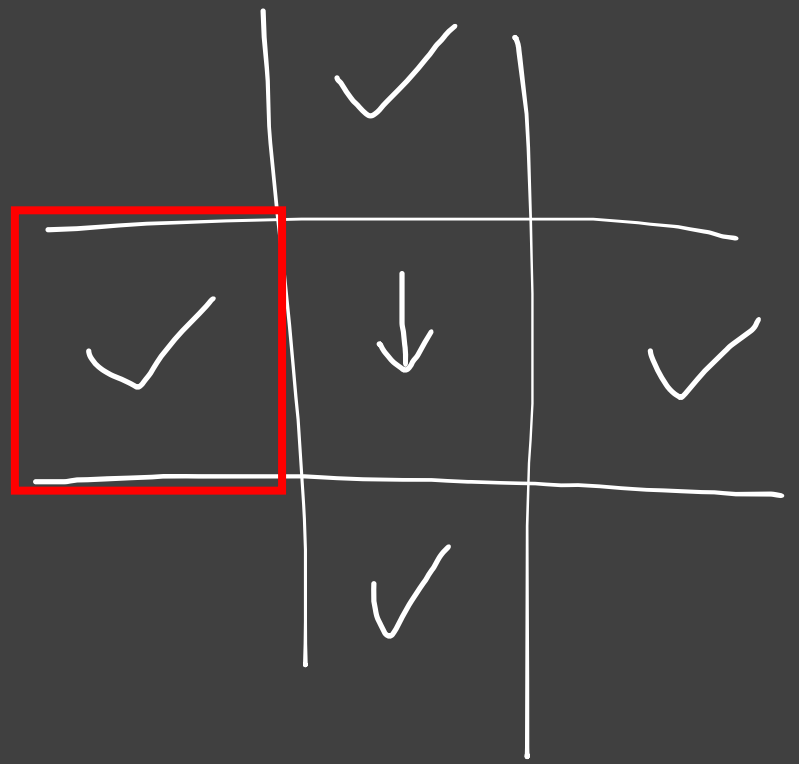
상하좌우 각각 배열 범위 안엔 들어가는지, 벽에 가로막히는지 검사하기엔 if문이 너무 많아 더러워짐

= 디버깅할 때 매우 힘들다

```
int d_x[] = { -1,1,0,0 };
int d_y[] = { 0,0,-1,1 };
for (int i = 0; i < 4; i++) {
    int nextX = d_x[i] + xPos;
    int nextY = d_y[i] + yPos;
    if (nextX >= 0 && nextX < M && nextY >= 0 && nextY < N) {
        if (visited[nextY][nextX] == false && x[nextY][nextX] == 1) {
            DFS(nextY, nextX);
        }
    }
}
```

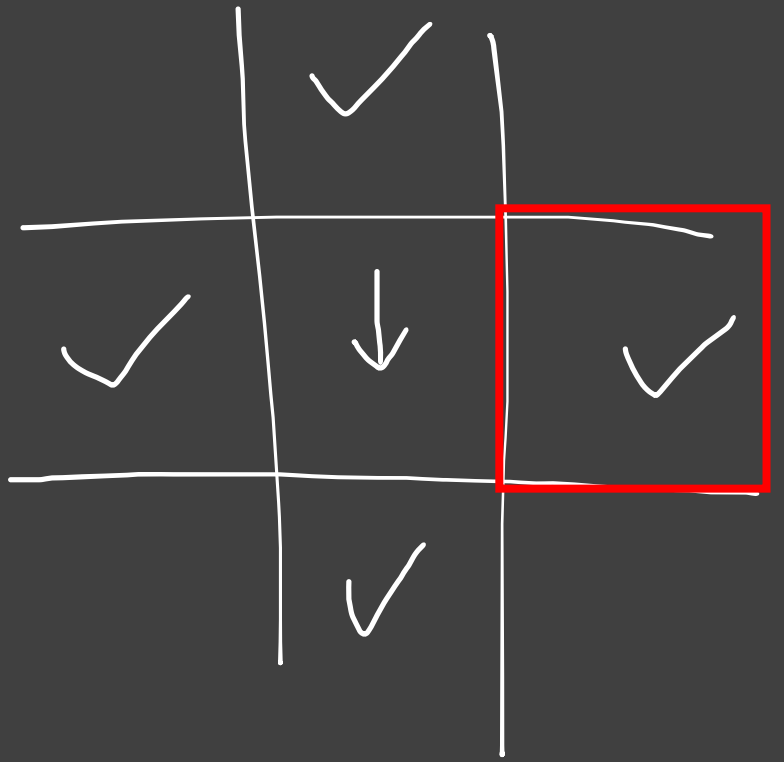
직접 코드를 디버깅해보자

```
int d_x[] = { -1,1,0,0 };
int d_y[] = { 0,0,-1,1 };
for (int i = 0; i < 4; i++) {
    int nextX = d_x[i] + xPos;
    int nextY = d_y[i] + yPos;
    if (nextX >= 0 && nextX < M && nextY >= 0 && nextY < N) {
        if (visited[nextY][nextX] == false && x[nextY][nextX] == 1) {
            DFS(nextY, nextX);
        }
    }
}
```



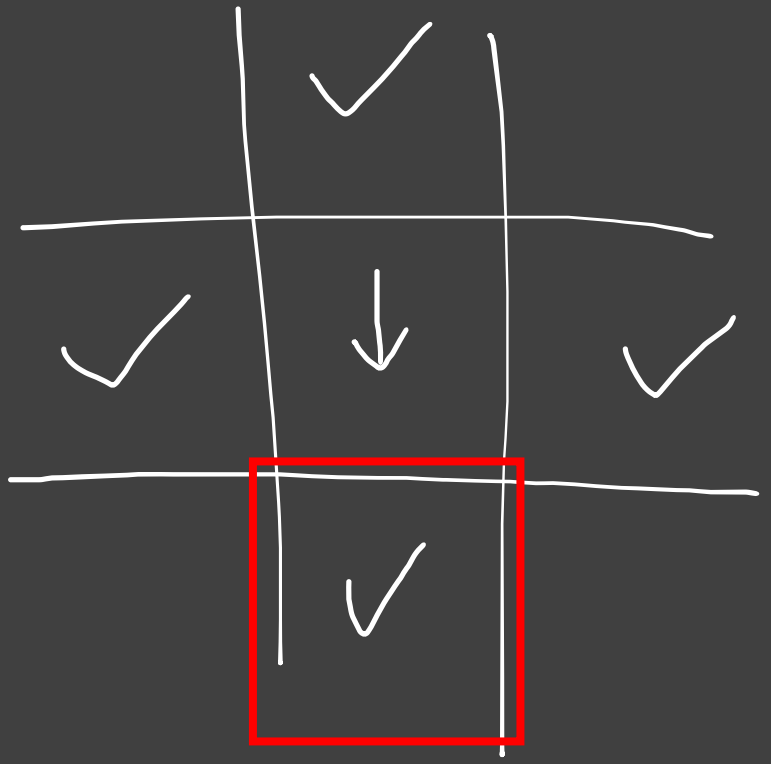
직접 코드를 디버깅해보자

```
int d_x[] = { -1,1,0,0 };
int d_y[] = { 0,0,-1,1 };
for (int i = 0; i < 4; i++) {
    int nextX = d_x[i] + xPos;
    int nextY = d_y[i] + yPos;
    if (nextX >= 0 && nextX < M && nextY >= 0 && nextY < N) {
        if (visited[nextY][nextX] == false && x[nextY][nextX] == 1) {
            DFS(nextY, nextX);
        }
    }
}
```



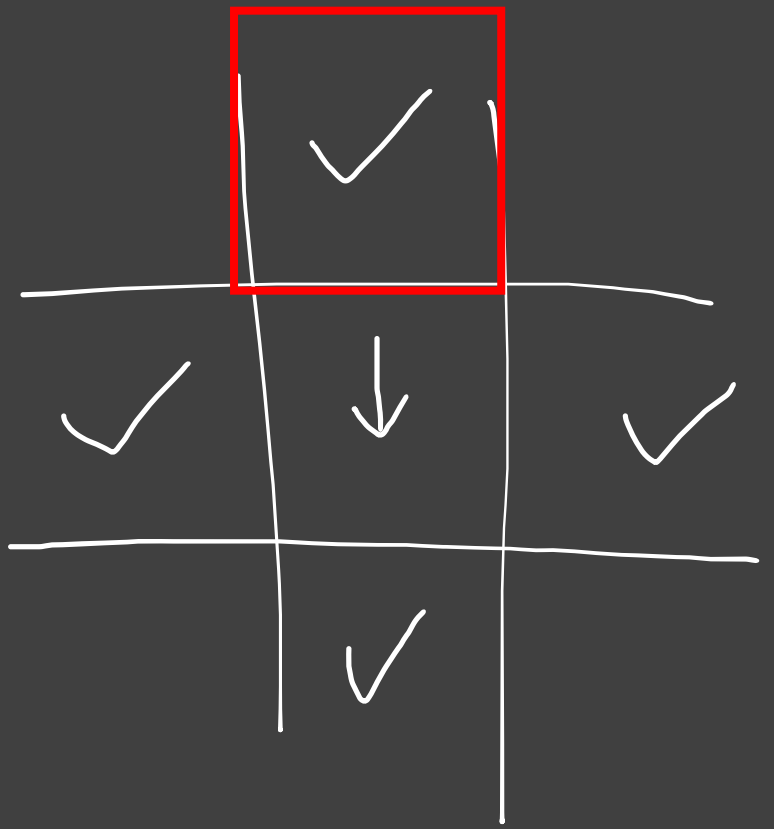
직접 코드를 디버깅해보자

```
int d_x[] = { -1,1,0,0 };
int d_y[] = { 0,0,-1,1 };
for (int i = 0; i < 4; i++) {
    int nextX = d_x[i] + xPos;
    int nextY = d_y[i] + yPos;
    if (nextX >= 0 && nextX < M && nextY >= 0 && nextY < N) {
        if (visited[nextY][nextX] == false && x[nextY][nextX] == 1) {
            DFS(nextY, nextX);
        }
    }
}
```



직접 코드를 디버깅해보자

```
int d_x[] = { -1,1,0,0 };
int d_y[] = { 0,0,-1,1 };
for (int i = 0; i < 4; i++) {
    int nextX = d_x[i] + xPos;
    int nextY = d_y[i] + yPos;
    if (nextX >= 0 && nextX < M && nextY >= 0 && nextY < N) {
        if (visited[nextY][nextX] == false && x[nextY][nextX] == 1) {
            DFS(nextY, nextX);
        }
    }
}
```



코드가 간단해진다!

연습 문제 추천

1012 – 유기농 배추 (Silver II)

10026 – 적록색약 (Gold V)

7576 – 토마토 (Gold V)

7569 – 토마토 (Gold V)

여러분들은 3번의 수업에 걸쳐 이러한 알고리즘을 배웠다

시간복잡도 이론, STL, queue, stack, string, greedy, 재귀 함수, Dynamic Programming, DFS, BFS 등등

2학기 수업 미리 스포

지금까지 배웠던 알고리즘을 활용한 더 다양한 알고리즘 선택, 조사, 발표

- 트리의 순회
- 다익스트라
- 유전 알고리즘
- 백트래킹
- 비트마스킹
- 세그먼트 트리
- 등등 여러분이 원하는 알고리즘도 가능

1학기동안 수고하셨습니다

마지막으로 생기부에 쓸 동아리 활동보고서 적고 디코에 제출해주세요