

SCV/2021

스타트 링크

- BFS -

안태진(taejin7824@gmail.com)

GitHub(github.com/taejin1221)

상명대학교 소프트웨어학과

201821002

Contents

- Problem
- Approach
- Code

Contents

- Problem
- Approach
- Code

Problem

• 문제 설명

문제

강호는 코딩 교육을 하는 스타트업 [스타트링크](#)에 지원했다. 오늘은 강호의 면접날이다. 하지만, 늦잠을 잔 강호는 스타트링크가 있는 건물에 늦게 도착하고 말았다.

스타트링크는 총 F층으로 이루어진 고층 건물에 사무실이 있고, 스타트링크가 있는 곳의 위치는 G층이다. 강호가 지금 있는 곳은 S층이고, 이제 엘리베이터를 타고 G층으로 이동하려고 한다.

보통 엘리베이터에는 어떤 층으로 이동할 수 있는 버튼이 있지만, 강호가 탄 엘리베이터는 버튼이 2개밖에 없다. U버튼은 위로 U층을 가는 버튼, D버튼은 아래로 D층을 가는 버튼이다. (만약, U층 위, 또는 D층 아래에 해당하는 층이 없을 때는, 엘리베이터는 움직이지 않는다)

강호가 G층에 도착하려면, 버튼을 적어도 몇 번 눌러야 하는지 구하는 프로그램을 작성하시오. 만약, 엘리베이터를 이용해서 G층에 갈 수 없다면, "use the stairs"를 출력한다.

• 스타트링크란?

- 백준이 만든 회사
- 코딩 교육도 하는 듯?

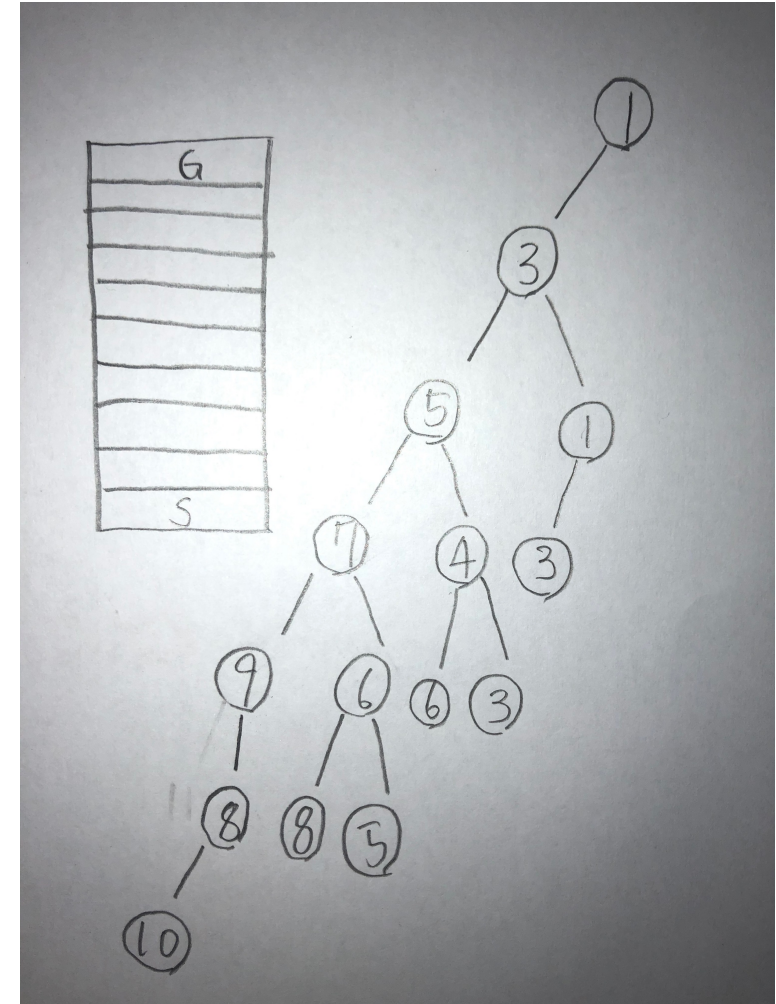


Contents

- Problem
- Approach
- Code

Approach

- 접근 방법
 - 시작 층에서 시작하여 갈 수 있는 모든 층을 트리로 그려봄
 - 오른쪽 그림의 트리를 탐색
 - 1에서 10까지의 최단 경로
 - 가중치가 없는 그래프에서는 BFS가 최단경로!
 - BFS 돌려 돌려!



Contents

- Problem
- Approach
- Code

Code

- Input

- 변수 이름 헛갈려서 바꿔버림

```
int maxFloor, startFloor, target, up, down;  
cin >> maxFloor >> startFloor >> target >> up >> down;
```

- 사용할 변수들 선언

- visited: 층이 방문한 적이 있는지
- ans: answer 즉, 답
- queue: BFS를 하기 위한 Queue
- pair<int, int>: { 층, 몇 초에 방문 했는지 } 의 쌍으로 Queue에 저장

```
bool visited[1'000'001] = { false, };  
  
int ans = -1;  
queue<pair<int, int>> q;
```


Code

- Queue의 초기값 설정
 - 시작점부터 시작이니 이미 방문을 함
 - 시작점은 0초에 방문

```
visited[startFloor] = true;  
q.push( { startFloor, 0 } );
```

- while 초기 설정
 - q가 비어있으면 모두 방문한 것
 - curr: 현재 노드
 - 만약 도착했다면 지금의 초를 답에 저장

```
while ( !q.empty() ) {  
    pair<int, int> curr = q.front(); q.pop();  
    if ( curr.first == target ) {  
        ans = curr.second;  
        break;  
    }  
}
```

Code

- Up, Down 버튼 누르기
 - 위층을 방문 안했고, 꼭대기가 아니라면
 - 1초 뒤에 방문!
 - 아래층을 방문 안했고, 1층 이하가 아니라면
 - 1초 뒤에 방문!
- 답 출력
 - ans의 초깃값은 -1, 따라서 while을 나온 뒤에도 -1이라면 못 가는 것임

```
int upFloor = curr.first + up, downFloor = curr.first - down;
if ( upFloor <= maxFloor && !visited[upFloor] ) {
    visited[upFloor] = true;
    q.push( { upFloor, curr.second + 1 } );
}

if ( 0 < downFloor && !visited[downFloor] ) {
    visited[downFloor] = true;
    q.push( { downFloor, curr.second + 1 } );
}
```

```
if ( ans == -1 )
    cout << "use the stairs";
else
    cout << ans;
cout << '\n';
```

감사합니다!
