

StartLink

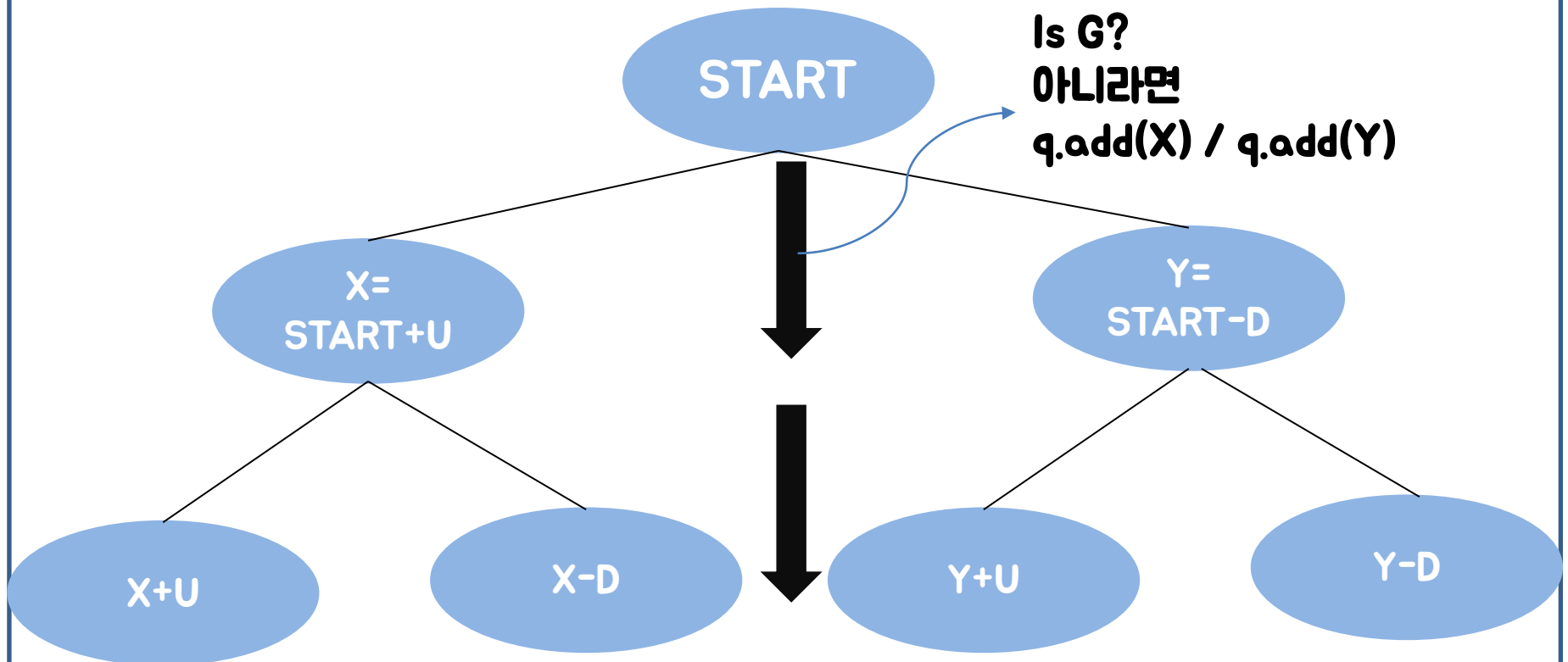


# BOJ 5014 풀이

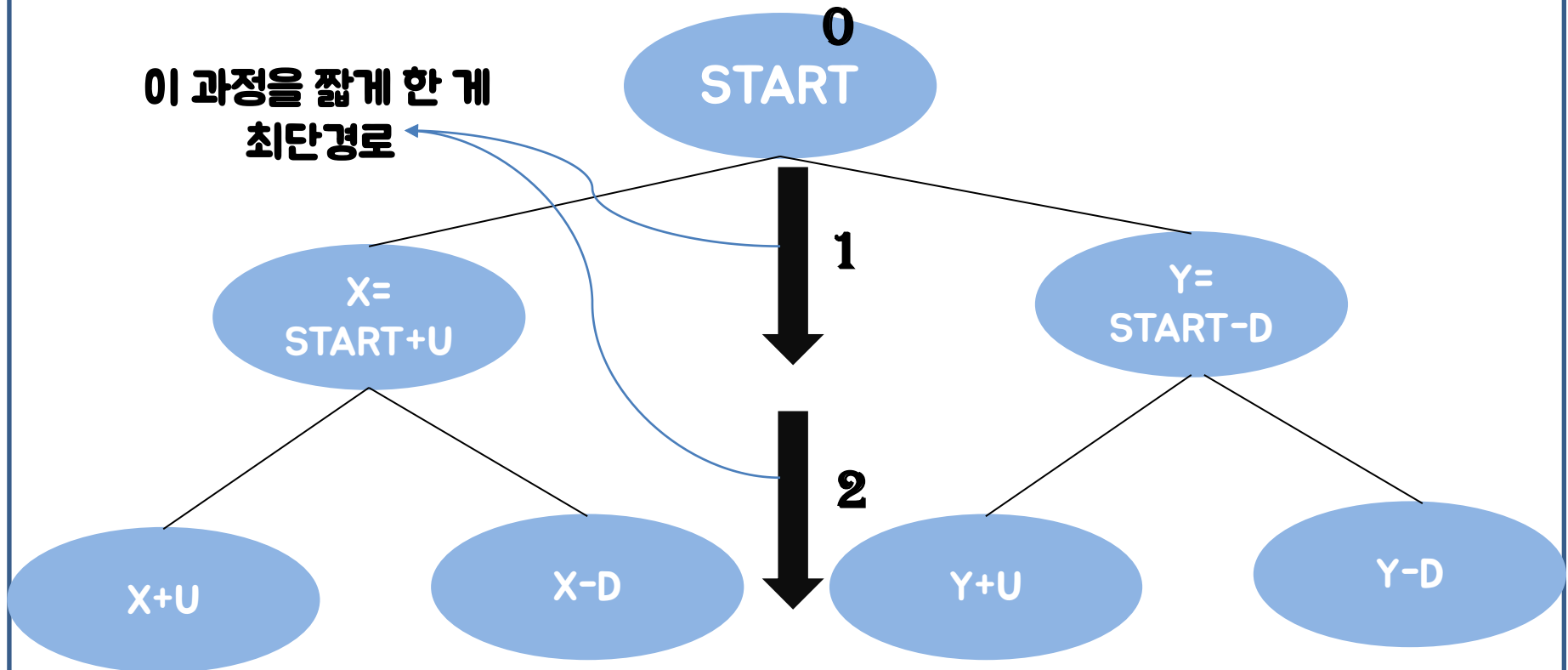
스타트링크

# **Approach**

## 시작점에서부터 U/D 한 것들 방문



아래의 과정을 가장 짧게 → **최단경로** (너비우선탐색)



**CODE**

# Main()

```
F=Integer.parseInt(st.nextToken());
S=Integer.parseInt(st.nextToken());
G=Integer.parseInt(st.nextToken());
U=Integer.parseInt(st.nextToken());
D=Integer.parseInt(st.nextToken());

arr=new int[F+1];
q.add(S);
arr[S]=1;

int result=BFS();
if(result==-1) bw.append("use the stairs");
else bw.append(result+"");
```

**arr[]** : 층별로 이동한 수 누적 (앞페이지의 화살표 개수) → +방문여부  
**result** : -1 or 최단경로  
**q** : BFS 활용할 Queue.

# BFS()

```
public static int BFS() {  
    while(!q.isEmpty()) {  
        int curF=q.poll();  
        if(curF==G) return arr[curF]-1;  
  
        if(1<=curF+U && curF+U<=F) { //U한 층이 범위내 && 방문X  
            if(arr[curF+U]==0 ) {  
                arr[curF+U]=arr[curF]+1;  
                q.add(curF+U);  
            }  
        }  
  
        if(1<=curF-D && curF-D<=F) { //D한 층이 범위내 && 방문X  
            if(arr[curF-D]==0 ) {  
                arr[curF-D]=arr[curF]+1;  
                q.add(curF-D);  
            }  
        }  
    }  
    return -1;  
}
```

1. 현재층을 꺼냄(직전에 넣었을 것)
2. 목적지 도착( $curF==G$ )이면  $arr[curF]-1$
3. 아직 멀었다면 U/D, Queue가 빌 때까지
4. Queue가 비었는데 도착X → 계단으로 가야 함