

**BOJ 25139**

Twitch Plays Vlllbit Explorer

# Step1. 접근

이동횟수는 상관 없으니,  
될 수 있는 대로 다 주워라!

- 현재 위치에서 갈 수 있는 아무데나 간다!
- 만약 그 위치에 주워야 하는 알파벳이 있다면 줍는다
- 아니라면 또 다시 갈 수 있는 곳으로 간다!

이동횟수는 상관이 없으나  
끝나는 지점은 정해져 있다  
아무렇게나 돌아다닐건데,  
언제 그 지점에 멈춰야 하는지 알 수 있을까?

# Step1. 접근

더 이상 남아 있는 알파벳으로는 강화를 할 수 없다는 것을 어떻게 알 수 있을까?

- 심지어 탈출 위치에 도착했을 때, 아이템을 가지고 있어서는 안된다
- 즉, 강화를 위한  $s * c$  개의 아이템을 정확히 줌고서 탈출해야 한다

# Step1. 접근

## C를 구해보자!

- 아이디에 등장하는 알파벳  $i$ 의 개수  $\rightarrow X(i)$
- 필드에 등장하는 알파벳  $i$ 의 개수  $\rightarrow Y(i)$
  
- 아이디에 어떤 알파벳  $i$ 가  $X(i)$ 번 등장한다고 할 때,
- $C$ 번 강화하기 위해서는 필드 안에는  $i$ 가 **적어도**  $C * X(i)$  개 존재해야 한다.
- 즉,  $C * X(i) \leq Y(i)$ 를 만족해야 한다.
  
- 각 알파벳에 대한  $C$ 를 구하고, 이때  **$C$ 의 최소값이 바로 최대 강화 횟수가 된다.**

# Step1. 접근

a b c a

a a a b b b b c

v	a	z	b
b	c	a	b
b	a	z	p

# Step1. 접근

a b c a

v a z b

b c a b

b a z p

a a a b b b b c

a b c a

$$\begin{aligned}C(a) &= Y(a)/X(a) = 3/2 = 1 \\C(b) &= Y(b)/X(b) = 4/1 = 4 \\C(c) &= Y(c)/X(c) = 1/1 = 1\end{aligned}$$

$\text{Min}(C(a), C(b), C(c)) = 1$  이므로  
최대 강화 횟수는 1 임!

# Step1. 접근

더 이상 남아 있는 알파벳으로는 강화를 할 수 없다는 것을 어떻게 알 수 있을까?

- 이제는 C를 알았기 때문에 알 수 있다!
- 아이디를 C번 완성하면 더 이상 강화할 수 없다는 것을 의미하기 때문이다!
- 즉, 아이디를 C번 완성하면 오른쪽 하단으로 이동하여 탈출하면 된다

# Step1. 접근

- 좌표 (0, 0) 에서 시작
- 현재 좌표로부터 이번 순서의 알파벳을 주우러 가기!
- 가는 길은 따로(command) 저장해두고
- 알파벳을 찾았다면 찾은 좌표 반환!
- 알파벳을 주운 자리임을 표시(\*)하고
- 다음 순서의 알파벳을 주우러 가기!



## Step2. 구현 – main()

```
int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);

    // 아이디에 등장하는 알파벳 i의 개수 -> X(i)
    vector<int> idAlphabet(NUMBER_OF_ALPHABET, 0);

    // 필드에 등장하는 알파벳 i의 개수 -> Y(i)
    vector<int> fieldAlphabet(NUMBER_OF_ALPHABET, 0);

    int S;
    cin >> N >> M >> S;

    // 필드 입력 및 필드에서의 각 알파벳 개수(Y) 카운트
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            cin >> field[i][j];
            fieldAlphabet[field[i][j] - 'a']++;
        }
    }
}
```

## Step2. 구현 – main()

// 아이디 입력

string uid;

cin >> uid;

// 아이디에서의 각 알파벳 개수(x) 카운트

```
for (int i = 0; i < S; i++) {  
    idAlphabet[uid[i] - 'a']++;  
}
```

// 최대 강화 횟수(C) 구하기

~~int C = INT\_MAX;~~

```
for (int i = 0; i < NUMBER_OF_ALPHABET; i++) {  
    if (idAlphabet[i] == 0) {  
        continue;  
    }  
}
```

```
C = min(C, fieldAlphabet[i] / idAlphabet[i]);  
}
```

## Step2. 구현 – main()

```
// 초기 위치 (0, 0)
pii curr = { 0, 0 };

// S * C 길이만큼 알파벳을 주워야 함
for (int i = 0; i < S * C; i++) {
    char ch = uid[i % S];
    curr = find(ch, curr);

    field[curr.first][curr.second] = '*'; // 알파벳 줄기
}

goToExit(curr);
cout << C << " " << command.size() << endl;
cout << command << endl;

return 0;
}
```

Handwritten notes:  $S * C$  (next to the for loop) and a bracket grouping the last two cout statements.

## Step2. 구현

goToExit()

C번의 강화 완료 후 오른쪽 하단으로 이동!

isValidPos()

이동할 좌표가 유효한 좌표인지 확인!

```
/* 오른쪽 하단으로 이동하는 함수 */  
void goToExit(pii pos) {  
    for (int i = 0; i < N - pos.first - 1; i++) {  
        command += "D";  
    }  
  
    for (int i = 0; i < M - pos.second - 1; i++) {  
        command += "R";  
    }  
}  
  
bool isValidPos(pii pos) {  
    return (pos.first < 0 || pos.first >= N || pos.second < 0 || pos.second >= M);  
}
```

## Step2. 구현

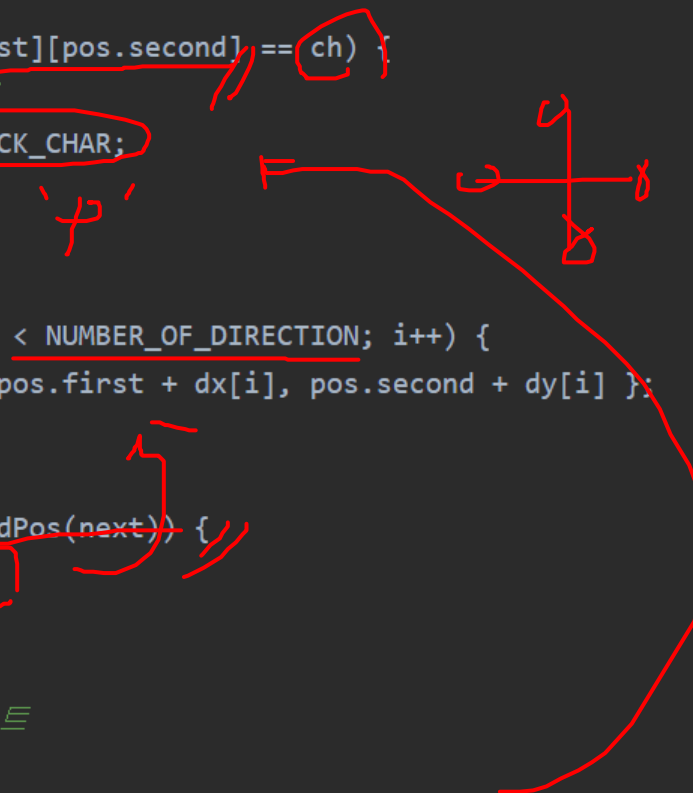
find()

이번에 주워야 하는 알파벳 ch를 찾기!

이동할 때마다 이동해온 방향을 저장.

주워야 하는 알파벳을 찾으면 해당 위치 반환하며 종료

```
pii find(char ch, pii pos) {  
    while (true) {  
        if (field[pos.first][pos.second] == ch) {  
            // 명령어 기록  
            command += PICK_CHAR;  
            break;  
        }  
  
        for (int i = 0; i < NUMBER_OF_DIRECTION; i++) {  
            pii next = { pos.first + dx[i], pos.second + dy[i] };  
  
            // 범위 확인  
            if (isValidPos(next)) {  
                continue;  
            }  
  
            // 좌표 업데이트  
            pos = next;  
            // 명령어 기록  
            command += dirName[i];  
            break;  
        }  
    }  
  
    return pos;  
}
```



# Step3. 결과

아이디	문제	결과	메모리	시간	언어
january	5 25319	메모리 초과			C++17 / 수정
january	5 25319	메모리 초과			C++17 / 수정
january	5 25319	런타임 에러 (InvalidPointer)			C++17 / 수정
january	5 25319	런타임 에러 (InvalidPointer)			C++17 / 수정
january	5 25319	런타임 에러 (OutOfBounds)			C++17 / 수정