

- 문제
 - 선행 정해져있는 것이 있다. (순서 맞춰야함). \Rightarrow 순서 주어진다.
 - ex) $A \rightarrow B \rightarrow C$ 이면, C를 배우기 전에 $A \rightarrow B$ 를 거쳐야함.
 - 정해지지 않은 것은 순서 X.
 - 유저가 만든 스킬트리중에서 가능한 것의 개수 구하는 문제.
 - Input : 스킬 순서 skill (길이는 20이상 26이하).
스킬트리는 `vector<string> skill-tree (1~20)`.

- 접근. skill = "CBD" 이고
 st[0] = "CBADF" 일 때.

① 해당 스킬트리의 문자가 skill에 있는 문자인지 확인해야함.

 < 없는 것이면 pass.

 < 있으면 { 해당 인덱스와 같은지 비교 (0차례).
 같다면 pass, 0 \rightarrow 1로 증가.
 (스킬 순서에 포함될 문자일 때만 cur을 비교, 증가).

② 위를 반복.

- 구현.

```
int solve (string s, vector<string> v) {
    int ans = 0;
    int check['Z'+1] = {0};
    for (int i=0; i<s.length(); ++i)
        check[s[i]] = i+1;
    for (int i=0; i<v.size(); ++i) {
        int cur = 1;
        bool is-right = true;
        for (int j=0; j<v[i].length(); ++j) {
            if (check[v[i][j]] == 0) continue;
            if (check[v[i][j]] == cur) cur++;
        }
    }
}
```

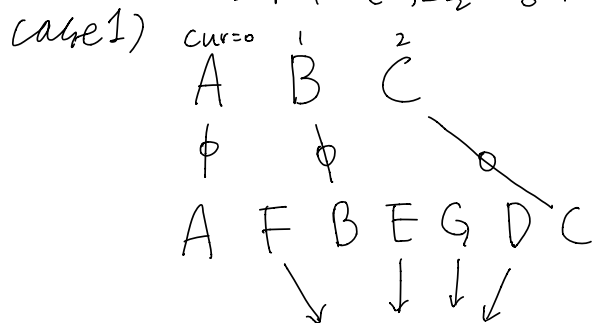
```

else {
    is_right = false;
    break;
}
}
if(is_right) ans++;
}
return ans;
}

```

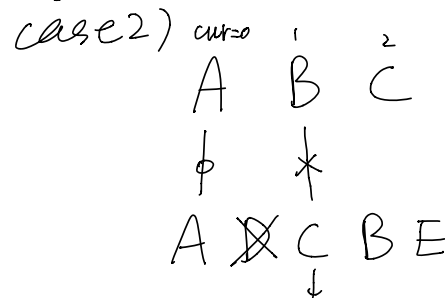
- 정리 : S와 V를 비교할때 V내에 포함된 S의 요소들이
 순서가 바르게 정렬되어있는지 알고싶을 때,
 S의 [0]~[len-1]까지의 문자에 대한 순서를 저장하기 위해
 char[26] or char['Z'] or map<char, int> 중
 하나를 미리 만들고.

임의의 인덱스 변수를 통해 S에 포함된 요소끼리만 순서를
 비교하며 인덱스를 증가시키면 된다.



S에 없으면 pass 하고

S에 있으면 cur 번째 스킵인지 체크해서, 아니면 (false; break;)
 맞다면 cur++;



C는 2번째 아직 cur=1.