

과제 #10 주사위 - 201921438 조현태

1) 소스코드

```
#include <stdio.h>
#include <iostream>
#include <string>
#include <string.h>
#include <algorithm>
#include <vector>
#include <math.h>
#include <stdlib.h>
#include <map>
#include <stack>
#include <queue>
using namespace std;

// 1 <= n <= 10
int n;

vector <string> dice; // 주사위 문자열 (6개짜리)
string answer[2401]; // 주사위로 만들 수 있는 모든 경우를 저장하는 문자열 (7x7x7x7)
vector <string> problems; // n개의 문제(문자열)을 저장하는 문자열

int main()
{
    cin >> n;

    // 주사위 문자열 4개로 초기화
    dice.clear(), dice.resize(4);

    // 주사위 문자열 입력
    for (int i = 0; i < 4; i++)
    {
        string s;
        cin >> s;
        // 마지막 원소에 0을 추가해서 주사위를 안쓰는 경우를 포함함.
        dice[i] = s + "0";
    }

    int a = 0;
    // 주사위로 나올 수 있는 모든 경우
```

```

for (int i = 0; i < 7; i++)
    for (int j = 0; j < 7; j++)
        for (int k = 0; k < 7; k++)
            for (int l = 0; l < 7; l++)
            {
                string s1, s2, s3, s4;
                s1 = dice[0].at(i);
                s2 = dice[1].at(j);
                s3 = dice[2].at(k);
                s4 = dice[3].at(l);

                answer[a] = s1 + s2 + s3 + s4;
                a++;
            }

```

```

// 문제 문자열 n개로 초기화
problems.clear(), problems.resize(n);

```

```

// 문제 문자열 저장
for (int i = 0; i < n; i++)
{
    string problem;
    cin >> problem;
    problems[i] = problem;
}

```

```

int no: // "NO"를 출력해도 되는지 판단
// n번(문제)만큼 반복
for (int i = 0; i < n; i++)
{
    // 모든 경우의 수에 대한 방문 판단 배열
    bool visited[2401][4] = {false, };

    // 현재 문제 문자열 설정
    string problem_n = problems[i];

    // 모든 경우에 대해서 실행
    for (int j = 0; j < size(answer); j++)
    {
        // 카운트 변수
        int count = 0;
        // NO변수는 1로 시작

```

```

no = 1;
// 문제 문자열만큼 반복
for (int k = 0; k < problem_n.size(); k++)
    // answer의 모든 원소에 접근
    for (int l = 0; l < 4; l++)
        // 방문하지 않아야 하고
        if (! visited[j][l])
            // 문제 문자열의 원소와 같아야 함
            if (problem_n[k] == answer[j][l])
            {
                // 방문처리
                visited[j][l] = true;
                // 카운트 증가
                count++;
                // 종료
                break;
            }
        // for문 종료 후 count가 문제 문자열의 길이와 같다면 -> 문제 문자열과 answer의 문자열이 같다.
        if (count == problem_n.size())
        {
            // YES 출력
            cout << "YES" << endl;
            // NO변수 0으로 변경
            no = 0;
            // 종료
            break;
        }
    }
// NO변수가 1이라면 NO출력
if (no == 1)
    cout << "NO" << endl;
}

return 0;
}

```

2) 문제 설명

주사위가 4개 주어지고 각 6개의 면에 임의의 문자가 있을때,
영문단어가 주어질 때, 만들 수 있는지를 판단하는 문제입니다.

n이 10까지이므로 시간 걱정을 하지 않고 완전 탐색으로 구현하였습니다.

주사위 4개로 1~4자리의 문자열을 만들 수 있는 경우는

$7 \times 7 \times 7 \times 7 = 2401$ 입니다. (6개의 면 + 주사위를 던지지 않는 경우)

즉, 2401개의 문자열이 생기는 것입니다.

주사위의 자리가 정해져 있는 것이 아니므로 영문단어의 문자가 모두 들어가면 정답입니다.

따라서 2401개의 문자열에 1~4의 문자를 모두 비교했습니다.

$2401(\text{모든 경우의 수}) \times 4!(\text{4자리 영문단어와 4자리 문자열을 문자 하나씩 비교할 경우})$

$\times 10(n\text{의 최대크기}) = 60\text{만정도라고 생각해서 바로 구현했습니다.}$

주사위가 4개이므로 4중 for문으로 만들 수 있는 모든 문자열을 만들고

모든 문자열의 모든 문자 하나하나와 영문단어 문자열의 모든 문자 하나하나와 비교하면서

같을 것이 있으면 종료하고 끝까지 탐색을 마치면 없는 것으로 판단하게 했습니다.

위의 과정을 n번 반복하면 제가 만든 알고리즘입니다.