백준 2482번 - 색상환

시간제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1초	512MB	8330	2874	2023	34.306%

문제

색을 표현하는 기본 요소를 이용하여 표시할 수 있는 모든 색 중에서 대표적인 색을 고리 모양으로 연결하여 나타낸 것을 색상환이라고 한다. 미국의 화가 먼셀(Munsell)이 교육용으로 고안한 20색상환이 널리 알려져 있다. 아래 그림은 먼셀의 20색상환을 보여준다.

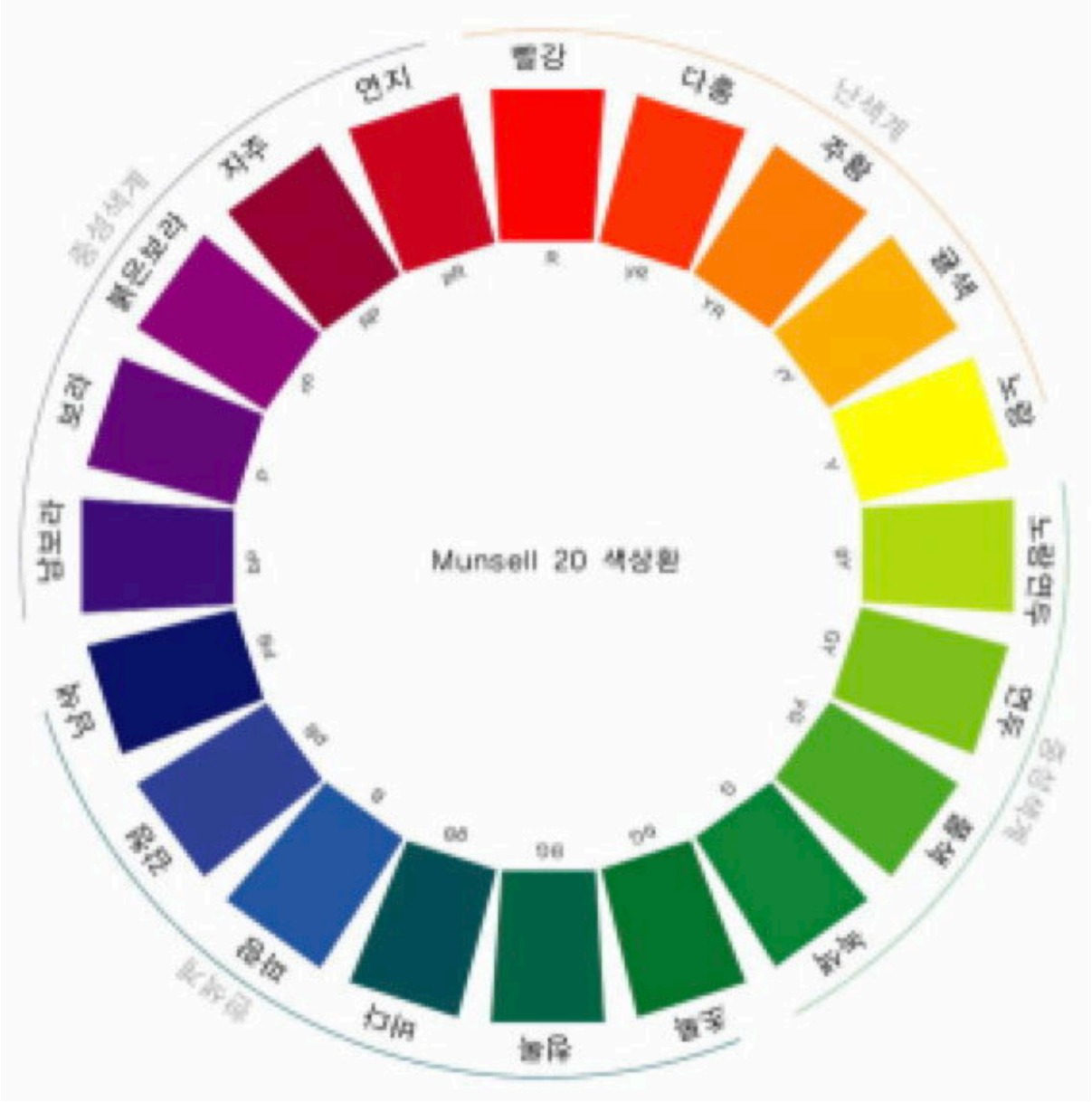

그림 1. 먼셀의 20색상환

색상환에서 인접한 두 색은 비슷하여 언뜻 보면 구별하기 어렵다. 위 그림의 20색상환에서 다홍은 빨강과 인접하고 또 주황과도 인접하다. 풀색은 연두, 녹색과 인접하다. 시각적 대비 효과를 얻기 위하여 인접한 두 색을 동시에 사용하지 않기로 한다.

주어진 색상환에서 시각적 대비 효과를 얻기 위하여 서로 이웃하지 않은 색들을 선택하는 경우의 수를 생각해 보자. 먼셀의 20색상환에서 시각적 대비 효과를 얻을 수 있게 10개의 색을 선택하는 경우의 수는 2이지만, 시각적 대비 효과를 얻을 수 있게 11개 이상의 색을 선택할 수 없으므로 이 경우의 수는 0이다.

주어진 정수 N과 K에 대하여, N개의 색으로 구성되어 있는 색상환 (N색상환)에서 어떤 인접한 두 색도 동시에 선택하지 않으면서 서로 다른 K개의 색을 선택하는 경우의 수를 구하는 프로그램을 작성하시오.

입력

입력 파일의 첫째 줄에 색상환에 포함된 색의 개수를 나타내는 양의 정수 N($4 \le N \le 1,000$)이 주어지고, 둘째 줄에 N색상환에서 선택할 색의 개수 K($1 \le K \le N$)가 주어진다.

출력

첫째 줄에 N색상환에서 어떤 인접한 두 색도 동시에 선택하지 않고 K개의 색을 고를 수 있는 경우의 수를 1,000,000,003 (10억 3) 으로 나눈 나머지를 출력한다

예제 입력1

```
4
2
```

예제 출력1

```
2
```

출처

출처

알고리즘 분류

- 다이나믹 프로그래밍

접근 방법

문제를 쉽게 해결하기 위하여 주어진 원순열에 위치를 고정시켜 선형적으로 표현해보자.

그렇다면, [$a_1a_2a_3...a_n$]와 같이 띠의 형태를 보이는데 우리는 실제 원형으로 이루어진 순열에 대하여 조건을 따져주어야 하므로 첫번째 색(a_1)과 마지막 색(a_n)을 제외한 임의의 위치(i) 에서 색을 선택할 경우 $i+1$ 위치에서는 색을 선택하지 않는 것을 통하여 `인접한 두 색을 선택하지 않는 경우의 수`를 따질 수 있게된다.

첫번째 색(a_1)과 마지막 색(a_n)은 위의 특징과는 조금 다른 양상을 띠는데, 이것들을 각각 경우의 수로 구분하여 각각의 경우에 대하여 문제를 해결해보자.

인접한 두 색을 선택하지 않아야하므로, a_1과 a_n 의 가능한 조합은 (0, 0), (0, 1), (1, 0) 총 3가지가 된다.

관찰을 통해, 임의의 위치 i에서 β개를 선택하는 모든 경우의 수는 (현재 위치i에서 색을 선택하는 경우) + (현재 위치 i에서 색을 선택하지 않는 경우)로 구분 될 수 있다.

1. 이 때, 색을 선택하는 경우는 이전 위치($i-1$)까지 선택한 색의 수가 $\beta-1$이여야 된다.
2. 색을 선택하지 않는 경우는 이전 위치($i-1$)까지 선택한 색의 수가 β 여야한다.

위에서 찾은 규칙성을 이용하기 위해 메모이제이션 테이블을 아래와 같이 정의할 수 있다.

`dp[i][j][0 or 1] := 현재 인덱스 i까지 j개의 색을 선택했을 때 가능한 모든 경우의 수(k : 단 현재 위치의 색을 뽑지 않았을 때, 뽑았을 때)`

이 때 첫번째 색(a_1)을 색칠하는 경우는 `dp[0][1][1] = 1` , 색칠하지 않는 경우는 `dp[0][0][0] = 1`으로 구분하여 base case를 잘 정의해준다.

$[a_2, a_{n-1}]$ 의 값들에 대해서는 위에서 기술한 조건에 따라 다음과 같은 점화식으로 표현할 수 있다.

$$dp[i][j][0] = dp[i-1][j][0] + dp[i-1][j][1]$$

$$dp[i][j][1] = dp[i-1][j-1][0]$$

a_n에 색을 칠할 경우 우리가 찾고자 하는 정답은 `dp[n-1][K][1]` 이 되고

a_n에 색을 칠하지 않는 경우 우리가 찾고자 하는 정답은 `dp[n-1][K][0]` 이 된다.

최종적으로, 첫번째 색과 마지막 색에 대해서 가능한 모든 경우의 수 (3가지)에 대해 위와 같이 경우의 수를 모두 더하는 것으로 우리가 원하는 정답을 구할 수 있다.

소스코드

```cpp
#define FASTIO cin.tie(0)->sync_with_stdio(false), cout.tie(0)
////////////////////////////////////////////////////////////////////
#include <bits/stdc++.h>
using namespace std;
const int MOD = 1e9 + 3;
int cand[3][2] = {{0, 0}, {0, 1}, {1, 0}};
int dp[1005][1005][2] = {0, };
int main(void){
    FASTIO;
////////////////////////////////////////////////////////////////////
    int N, K, ans = 0;
    cin >> N >> K;

    for(const auto &[fc, lc] : cand){
        memset(dp, 0, sizeof(dp));
        // [0, N-1] => 0, [1, N-2], N-1
        dp[0][0][0] = !fc;
        dp[0][1][1] = fc;

        for(int i=1; i<N-1; i++){
            for(int j=0; j<=K; j++){
                dp[i][j][0] = (dp[i-1][j][0] + dp[i-1][j][1]) % MOD;
                dp[i][j][1] = j-1 >= 0 ? dp[i-1][j-1][0] : 0;
            }
        }
        dp[N-1][K][0] = (dp[N-2][K][0] + dp[N-2][K][1]) % MOD;
        dp[N-1][K][1] = dp[N-2][K-1][0];

        ans += dp[N-1][K][lc];
        ans %= MOD;
```

```
    }
    cout << ans ;
    return 0;
}
```