



BOJ 2482 : 색상환

통계 수정 삭제

beberiche · 방금 전

 0

DP algorithms boj

algorithms



▼ 목록 보기

3/3



문제

색상환

성공



시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	128 MB	9801	3448	2434	34.345%

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 교육 용으로 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 N 과 K 에 대하여, N 개의 색으로 구성되어 있는 색상환 (N 색상환)에서 어떤 인접한 두 색도 동시에 선택하지 않으면서 서로 다른 K 개의 색을 선택하는 경우의 수를 구하는 프로그램을 작성하시오.

입력

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

출력

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

예제 입력 1 복사

4

2

예제 출력 1 복사

2

출처

Olympiad > 한국정보올림피아드 > 한국정보올림피아드시.도지역본선 > 지역본선 2010 > 중등부 4번

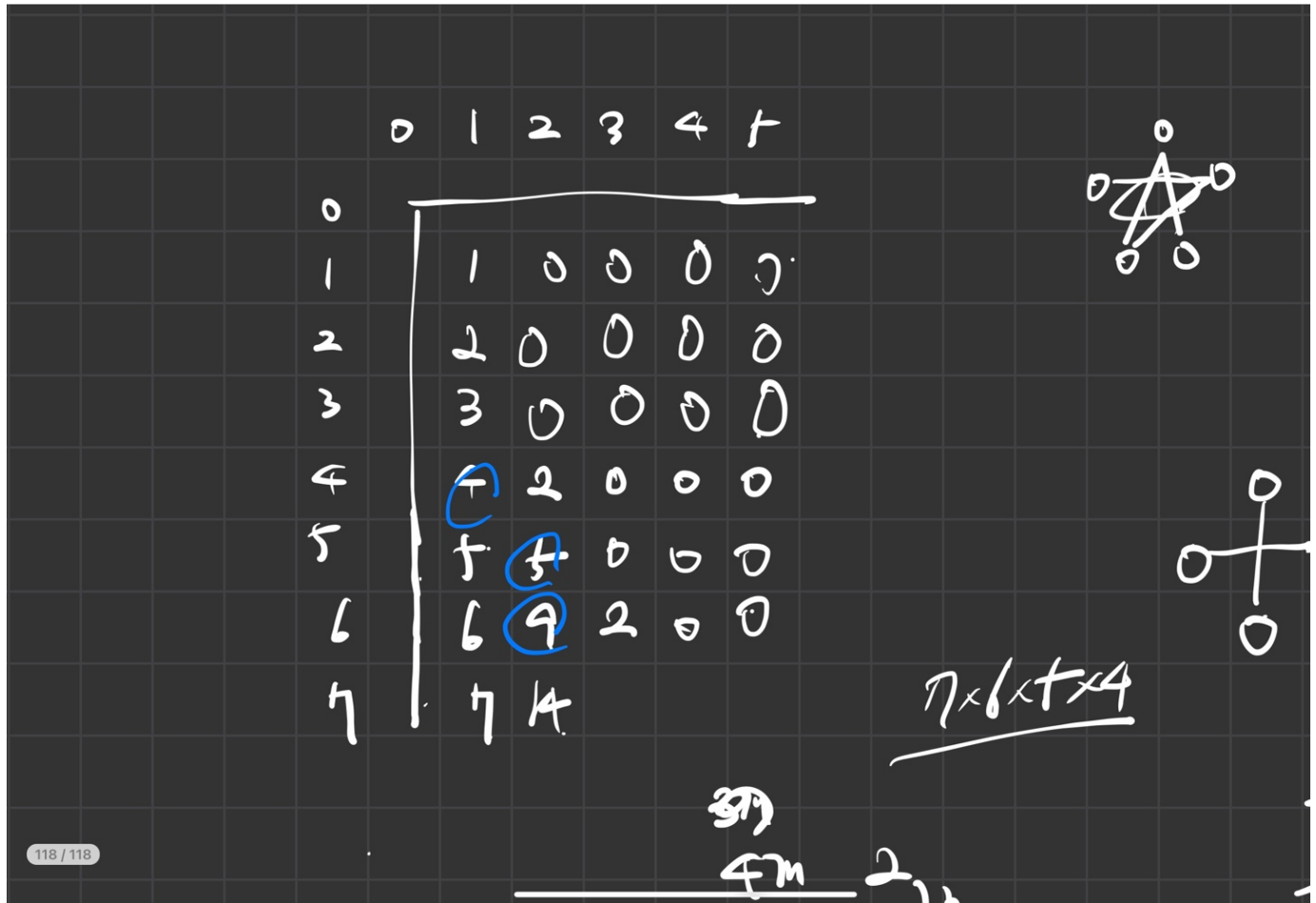
Olympiad > 한국정보올림피아드 > 한국정보올림피아드시.도지역본선 > 지역본선 2010 > 고등부 3번

- 문제의 오타를 찾은 사람: hihih
- 데이터를 추가한 사람: jeong5728, jinhan814

풀이과정

전형적인 DP 문제. 자신에 인접한 색을 제외한 K 개수 만큼의 경우의 수를 N 에서 찾아야한다. 처음엔 인접하다길래 그래프인가 싶었지만, $_nC_k$ 의 모든 경우를 탐색해야하므로, 완탐시 시간초과가 날 수 있다고 예상. 분명 누적하여 진행하는 공식이 있을거라고 추측했다.

수학이랑 친하지 않은 편이라, 늘 DP 문제에서는 점화식을 세우기보다는 일일이 작성해보며 규칙을 찾곤 한다.



규칙 1

N 은 색의 총 갯수, K 는 뽑아야 하는 가짓수라고 한다면, 일단 $K = 1$ 인 경우에는 자신만 뽑히니 N 가지의 경우의 수가 나온다.

규칙 2

$N/2$ 보다 K 는 클 수 없다. $N = 2$ 인데 3가지 경우의 수를 뽑을 순 없지 않은가? 또한 인접한 수는 제외되기 때문에, 자신과 인접한 2가지 수를 제외하면 N 은 적어도 4 이상이어야 한다.

점화식

이를 기반으로, 직접 그려가며 확인해본 결과 다음과 같은 규칙을 발견할 수 있었다.

$$DP[N][K] = DP[N-2][K-1] + DP[N-1][K]$$

오답

출력 값은 반드시 $1e9+3$ 을 나눈 나머지 값으로 나오게 하라고 하였다. $1e9+3$ 으로 나머지 값을 찾으니 double 값을 나타내게 되어서, 이를 int 화 시켰더니 다른 답이 나오게 되었다.

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();
        int K = sc.nextInt();

        int[][] dp = new int[N+1][K+1];

        for(int i=1; i<dp.length; i++) dp[i][1] = i;

        for(int i=4; i<=N; i++) {
            for(int j=2; j<=K; j++) {
                dp[i][j] = (dp[i-2][j-1] + dp[i-1][j]) % (int) 1e9+3;
            }
        }
        System.out.println(dp[N][K]);
    }
}
```

$N = 4, K = 2$ 를 가지고 디버깅을 진행해본 결과 $N = 4, K = 2$ 는 2 가 나와야 하는데, 5 의 값이 나왔다.

정답

$1e9+3$ 을 전체적으로 int 화 시키니 답이 제대로 나왔다. 어떤 부분이 다른 건지는 잘 모르겠다. 점화식을 세우는 것도 그렇지만, 이러한 부분에서 시간을 많이 뺏기는 것이 아쉬웠다.

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();
        int K = sc.nextInt();

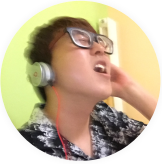
        int[][] dp = new int[N+1][K+1];

        for(int i=1; i<dp.length; i++) dp[i][1] = i;

        for(int i=4; i<=N; i++) {
            for(int j=2; j<=K; j++) {
```

```
        dp[i][j] = (dp[i-2][j-1] + dp[i-1][j]) % (int) (1e9+3);
    }
}
System.out.println(dp[N][K]);
}
}
```

언제쯤 수학과 친해지려나...



벨

새로운 것에 관심이 많고, 프로젝트 설계 및 최적화를 좋아합니다.



이전 포스트

BOJ 11051 : 이항계수2

0개의 댓글

댓글을 작성하세요

댓글 작성



Powered by
Stellate