



델리만주는 무서워!

ALPS 2020년 7월 내부대회 C번

출제자: 김민채

풀이

동적계획법으로 풀어야 하는 문제입니다. 문제 안에서 일정한 규칙을 찾아, 점화식을 도출해내는 것이 핵심입니다.

델리만जू 2개를 나누어 먹는 방법은 한 가지로, 다음과 같습니다.

- 1일 차: 2개

델리만जू 3개를 나누어 먹는 방법은 한 가지로, 다음과 같습니다.

- 1일 차: 3개

델리만जू 4개를 나누어 먹는 방법은 두 가지로, 다음과 같습니다.

- 1일 차: 4개
- 1일 차: 2개, 2일 차: 2개

각 경우에 대하여, 그 다음 날 먹을 수 있는 델리만जू의 수는 2개, 3개, 4개 입니다. 따라서 델리만जू 2개를 먹는 방법에서 다음 날 4개를 먹는다면, 총 델리만जू 6개를 먹는 방법이 될 것입니다. 델리만जू 3개, 4개를 먹는 방법에 대해서도 각각 다음 날 3개, 2개를 더 먹는다면 델리만जू 6개를 먹는 방법이 될 것입니다.

아래를 봅시다.

5개

[2개 + 3개]

- 1일 차: 2개, 2일 차: 3개

[3개 + 2개]

- 1일 차: 3개, 2일 차: 2개

6개

[2개 + 4개]

- 1일 차: 2개, 2일 차: 4개

[3개 + 3개]

- 1일 차: 3개, 2일 차: 3개

[4개 + 2개]

- 1일 차: 4개, 2일 차: 2개

- 1일 차: 2개, 2일 차: 2개, **3일차: 2개**

7개

[3개 + **4개**]

- 1일 차: 3개, **2일 차: 4개**

[4개 + **3개**]

- 1일 차: 4개, **2일 차: 3개**
- 1일 차: 2개, 2일 차: 2개, **3일 차: 3개**

[5개 + **2개**]

- 1일 차: 2개, 2일 차: 3개, **3일 차: 2개**
- 1일 차: 3개, 2일 차: 2개, **3일 차: 2개**

$dp[i]$ 는 델리만쥬 i 개를 먹는 방법의 수를 1,000,000,007로 나눈 나머지라고 합시다.

위를 통해,

$$dp[i] = (dp[i - 2] + dp[i - 3] + dp[i - 4]) \% 1,000,000,007 \quad (i \geq 5)$$

라는 점화식이 도출됨을 알 수 있습니다.

만약, 쿼리가 들어올 때마다 dp 테이블을 새로 채운다면 너무 많은 연산이 필요해 1000ms의 시간 제한을 지킬 수 없습니다. 현재 문제에서는 각 쿼리에 주어지는 델리만쥬 수가 달라지더라도 dp 테이블이 공유 가능하기 때문에, 앞선 쿼리에서 정답을 계산하기 위해 채운 dp 테이블을 그대로 사용해야 합니다.

가장 좋은 방법은, 쿼리를 입력받기 앞서 모든 dp 값을 미리 계산해 두는 것입니다.

이에 시간복잡도는 $O(Q + N)$ 입니다.

정해 코드

```
#include <iostream>

using namespace std;

#define MAX 200001

int mod = 1e9+7;
long long dp[MAX];

int main() {
    int q, n;

    dp[0] = 0;
    dp[1] = 0;
    dp[2] = 1;
    dp[3] = 1;
    dp[4] = 2;

    scanf("%d", &q);
```

```
for(int i=5; i<MAX;i++){
    dp[i] = (dp[i-2]+dp[i-3]+dp[i-4])%mod;
}

for(int i =0; i<q; i++){
    scanf("%d",&n);
    cout << dp[n] << "\n";
}

return 0;
}
```