

# Codeforces 223 C Partial Sums

Combinatorics

## Description

Given an array  $a[]$ , define an operation:

$$s_i = \sum_{j=1}^n a_j, s_i \rightarrow a_i.$$

Calculate the value of each  $a$  after  $k$  times of operations.

## Solution

The official tutorial wrote confusingly, so did other solutions on Internet.

Calculate how element  $a_j$  will affects  $a_i$  ( $j \leq i$ ), after  $k$  times.

Let's draw something first.

$(i, j, k)$  means  $i \cdot a_1 + j \cdot a_2 + k \cdot a_3$ , and  $K$  is times of operation :

$$\begin{cases} (1, 0, 0) (1, 1, 0) (1, 1, 1) & \text{where } K = 1, \\ (1, 0, 0) (2, 1, 0) (3, 2, 1) & \text{where } K = 2, \\ (1, 0, 0) (3, 1, 0) (6, 3, 1) & \text{where } K = 3, \\ \dots \end{cases}$$

We also have another graph:

```
1 1
1 2 1
1 3 3 1
1 4 6 4 1
...
```

Think when  $k$  is changing, how the factor before  $a_j$  in  $a_i$  follows, we get:

$$fac_{i,j}^k = \binom{j-i+k-1}{j-i}$$

To simplify our calculation, we found that  $fac_{i,j}^k = fac_{i-j+1,1}^k$ , for we can replace  $j-i$  with  $j-i+1-1$ , they are the same!

So our task now is to calculate  $fac_{i,1}^k$ , which is  $\binom{i+k-2}{i-1}$ . But we cannot pre process the factorial of  $k$ , as  $k$  can be up to  $10^9$ . But  $fac_{1,1}$  is always 1!

Think the transfer between  $fac_{i-1,1}$  and  $fac_{i,1}$ , that is:

$$\frac{(i-1+k-2)!}{(i-1-1)!(k-1)!} \rightarrow \frac{(i+k-2)!}{(i-1)!(k-1)!}$$

We just need to multiply the first expression by  $i+k-2$  and  $(i-1)^{-1} \pmod{10^9+7}$ . And we can pre process all factors in  $O(n)$  time.

Then finish all calculation in  $O(n^2)$  time, as  $s_i = \sum_{j=1}^i f_{i-j+1} \times a_j$ .

## Code

```
#include <iostream>
#include <algorithm>
#include <cstring>
#include <cstdio>
using namespace std;
typedef long long ll;
const ll p = 1e9 + 7;
const int maxn = 2005;
int n, k;
ll inv[maxn], a[maxn], c[maxn], s[maxn];
void init() {
    inv[1] = 1;
    for (int i = 2; i <= 2000; ++i) {
        inv[i] = -(p/i) + p * inv[p % i] % p;
    }
}
int main() {
    init();
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);
    cin >> n >> k;
    for (int i = 1; i <= n; ++i) cin >> a[i];

    if(!k) {
        for(int i = 1; i <= n; ++i) s[i] = a[i];
    } else {
        c[1] = 1;
        for (int i = 2; i <= n; ++i) c[i] = c[i - 1] * (k + i - 2) % p * inv[i - 1] % p;
        for (int i = 1; i <= n; ++i)
            for (int j = 1; j <= i; ++j) s[i] = (s[i] + c[i - j + 1] * a[j] % p) % p;
    }
    for (int i = 1; i <= n; ++i) cout << s[i] << " ";
    cout << endl;
    return 0;
}
```