

Flower Gathering

Problem Explanation

There are N flower plants planted in a row numbered from 1 to N . Each flower plant gives $A[i]$ flowers where $1 \leq i \leq N$. Sam has a special power using which he can multiply the flowers he gathers on a day with $(k - (\text{current day} - 1))$ but he can only use it at most k times. The numbering of the day starts from 1. We need to find the subsequence in which Sam gathers flows such that he has maximum flowers at the end.

Prerequisites

Dynamic Programming

Approach

What is a subsequence?

A **subsequence** is a **sequence** that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements.

Let $dp[i][j]$ be the maximum flowers you can get starting from the i^{th} index and j^{th} day. So at each recursive there are 2 options -

1. Multiply $a[i]$ with $(k+1-j)$ and add the value with the next recursive call where i and j increases by 1.
2. Do not include the current value and have the next recursive call where i increases by 1 and j remains the same.

So, $dp[i][j]$ will be the maximum of the above 2 steps.

Note, j should always be less than equal to k .

```
long long dp[1005][1005];
long long solve(int ind,int rem)
{
    if(rem==k+1 || ind==n+1) return 0;

    if(dp[ind][rem]!=-1) return dp[ind][rem];
    long long ans=0;
    ans=max(a[ind]*(k+1-rem)+solve(ind+1,rem+1),solve(ind+1,rem));
    dp[ind][rem]=ans;
    return ans;
}
```