629. K Inverse Pairs Array

- Difficulty:**Medium**
- Category:Algorithms
- Acceptance:20.61%
- Contributor:Stomach_ache

Given two integers `n` and `k`, find how many different arrays consist of numbers from `1` to `n` such that there are exactly `k` inverse pairs.

We define an inverse pair as following: For i_{th} and j_{th} element in the array, if `i` < `j` and `a[i]` > `a[j]` then it's an inverse pair; Otherwise, it's not.

Since the answer may very large, the answer should be modulo $10^9 + 7$.

Example 1:

```
Input: n = 3, k = 0

Output: 1

Explanation:

Only the array [1,2,3] which consists of numbers from 1 to 3 has exactly 0 inverse pair.
```

Example 2:

```
Input: n = 3, k = 1

Output: 2

Explanation:

The array [1,3,2] and [2,1,3] have exactly 1 inverse pair.
```

Note:

1. The integer `n` is in the range [1, 1000] and `k` is in the range [0, 1000].

dp[n][k] denotes the number of arrays that have k inverse pairs for array composed of 1 to n

we can establish the recursive relationship between dp[n][k] and dp[n-1][i]:

if we put n as the last number then all the k inverse pair should come from the first n-1 numbers

if we put n as the second last number then there's 1 inverse pair involves n so the rest k-1 comes from the first n-1 numbers

if we put n as the first number then there's n-1 inverse pairs involve n so the rest k-(n-1) comes from the first n-1 numbers

dp[n][k] = dp[n-1][k]+dp[n-1][k-1]+dp[n-1][k-2]+...+dp[n-1][k+1-n+1]+dp[n-1][k-n+1]

It's possible that some where in the right hand side the second array index become negative, since we cannot generate negative inverse pairs we just treat them as 0, but still leave the item there as a place holder.

dp[n][k] = dp[n-1][k]+dp[n-1][k-1]+dp[n-1][k-2]+...+dp[n-1][k+1-n+1]+dp[n-1][k-n+1] + (n-1)n/2

dp[n][k+1] = dp[n-1][k+1]+dp[n-1][k]+dp[n-1][k-1]+dp[n-1][k-2]+...+dp[n-1][k+1-n+1]+n(n-1)/2

so by deducting the first line from the second line, we have

dp[n][k+1] = dp[n][k]+dp[n-1][k+1]-dp[n-1][k+1-n]

//author:MR.BLACK

```
class Solution {

public:

    int kInversePairs(int n, int k) {

        int mod = 1000000007;

        //k>c_n^2

        if(k>n*(n-1)/2 || k<0) return 0;

        if(k==n*(n-1)/2 || k==0) return 1;

        vector<vector<long>> dp(n+1,vector<long>(k+1,0));

        dp[2][0] = 1;

        dp[2][1] = 1;

        for(int i=3;i<=n;i++)

        {
            dp[i][0]=1;
            for(int j=1;j<=min(i*(i-1)/2,k);j++)
            {
                dp[i][j] = dp[i][j-1] + dp[i-1][j];
                if(j>=i)
                {
                   dp[i][j] = dp[i][j] - dp[i-1][j-i];
                }
                dp[i][j] = (dp[i][j]+mod)%mod;
            }
        }
        return (int)dp[n][k];
    }
};
```