<https://www.hackerrank.com/contests/101hack18/challenges/candles-2>

## Candles Counting

**locked**

* [**Problem**](https://www.hackerrank.com/contests/101hack18/challenges/candles-2)
* [**Submissions**](https://www.hackerrank.com/contests/101hack18/challenges/candles-2/submissions)
* [**Leaderboard**](https://www.hackerrank.com/contests/101hack18/challenges/candles-2/leaderboard)
* [**Discussions**](https://www.hackerrank.com/contests/101hack18/challenges/candles-2/forum/questions)
* [**Editorial**](https://www.hackerrank.com/contests/101hack18/challenges/candles-2/editorial)

Tim is visiting his grandma for two days and is bored due to the lack of the electricity over there. That's why he starts to play with grandma's colorful candle collection.

He aligned the N candles from left to right. The ith candle from the left has the height Hiand the color Ci, an integer ranged from 1 to a given K, the number of colors.

Now he stares at the sequence of candles and wonders, how many strictly increasing ( in height ) colorful subsequences are there? A subsequence is considered as colorful if every of the K colors appears at least one times in the subsequence.

As the number of subsequences fulfilling the requirement can be large, print the result modulo 109+7.

**Input Format**

On the first line you will be given N and K, then N lines will follow. On the ith line you will be given two integers Hi and Ci.

**Output Format**

Print the number of strictly increasing colorful subsequences modulo 109+7.

**Constraints**

* 1≤N≤5⋅104
* 1≤Ci≤K≤7
* 1≤Hi≤5⋅104

**Sample Input**

4 3

1 1

3 2

2 2

4 3

**Sample Output**

2

**Explanation**

In the first sample the only two valid subsequences are (1, 2, 4) and (1, 3, 4).

[Suggest Edits](https://www.hackerrank.com/)

#include <bits/stdc++.h>

using namespace std;

#define pb push\_back

#define mp make\_pair

#define sz(s) ((int)(s.size()))

#define bg begin()

#define en end()

#define Y second

#define X first

typedef long long ll;

#define fi freopen("input.txt","r",stdin)

#define fo freopen("output.txt","w",stdout)

const double pi = acos(-1.0);

const double eps = 1e-8;

#define print(a) cout<<(#a)<<" = "<<a<<"\n";

#define fill(a,val) memset(a ,val, sizeof(a) );

/\*Solution code starts here \*/

const ll mod = 1000000007LL;

#define maxn 100009

ll tree[129][maxn];

int in[maxn],col[maxn];

void set\_Value( int mask , int id , ll val )

{

while( id < maxn )

{

tree[mask][id]=(tree[mask][id] + val )%mod;;

id+=( id & (-id));

}

}

ll get\_value( int mask , int id)

{

ll ans=0;

while( id > 0)

{

ans=(ans + tree[mask][id] )%mod;

id-=( id & (-id));

}

return ans;

}

int main()

{

ios\_base::sync\_with\_stdio(0);

fill(tree,0);

int N,K;

cin>>N>>K;

for(int i=1;i<=N;i++)

{cin>>in[i]>>col[i];

col[i]--;

}

int mx=0;

for(int i=1;i<=N;i++)

{

mx=max( mx , in[i] );

ll newval[130];

fill(newval,0);

for(int mask=0;mask<(1<<K);mask++)

{

ll tp=get\_value(mask, in[i]-1 );

newval[ mask | (1<<col[i]) ] = ( newval[ mask | (1<<col[i]) ] + tp ) % mod;;

}

newval[ (1<<col[i]) ] = ( newval[(1<<col[i]) ] + 1 ) % mod; //only this one

for(int mask = 0; mask<(1<<K); mask++ )

set\_Value( mask, in[i], newval[mask] );

}

cout<<get\_value((1<<K)-1,mx);

return 0;

}