Анализ на задача magic

Ограниченията на задачата подсказват, че тя може да бъде решена чрез симулиране на казаното в нея (дните, в които магът създава дракони). Тук трудността идва в това, че трябва да бъдат имплементирани дълги числа, защото отговорът може да достигне много големи стойности.

#include <cstdio>

#include <iostream>

#include <string>

using namespace std;

const int maxk = 128;

int N , K , a[maxk];

void read (){

 scanf ( "%d%d" , &N , &K );

 int d;

 for ( int i = 0; i < K ; ++i ) {

 scanf ( "%d" , &d );

 a[d] = 1;

 }

}

string add ( string s ) {

 int sz = s.size();

 int b[120004];

 for ( int i = 0; i < sz; ++i )

 b[i] = s[i] - '0';

 b[sz-1] ++;

 int p ;

 if ( b[sz-1] == 10 ) {

 b[sz-1] = 0;

 p = 1;

 } else p = 0;

 for ( int i = sz-2; i >= 0; --i ){

 if ( (b[i] + p )== 10 ) {

 b[i] = 0;

 p = 1;

 } else {

 b[i] += p;

 p = 0;

 }

 }

 string ret;

 if ( p == 1 ) ret = "1";

 else ret = "";

 for ( int i = 0; i < sz; ++i )

 ret += char(b[i] +'0' ) ;

 return ret;

}

string mul ( string s ) {

 int sz = s.size();

 int b[120004];

 for ( int i = 0; i < sz; ++i )

 b[i] = (s[i] - '0');

 int p = 0;

 for ( int i = sz-1; i >= 0; --i ){

 b[i] = b[i]\*2;

 b[i] += p;

 p = b[i] / 10;

 b[i] = b[i] % 10;

 }

 string ret="";

 if ( p ) ret += char(p+'0');

 else ret = "";

 for ( int i = 0; i < sz; ++i )

 ret += char(b[i] +'0' ) ;

 return ret;

}

void solve (){

 string ans = "0";

 for ( int i = 0; i < N; ++i ) {

 ans = add(ans);

 if ( a[i+1] ) ans = mul ( ans );

 }

 cout << ans << endl;

}

int main (){

 read ();

 solve ();

 return 0;

}