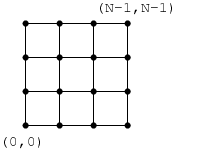
Préparation au concours ACM — TP4  
Christoph Dürr Jill-Jênn Vie  
9 octobre 2014

Codility —Psi2012

There is an N **×** N square mesh-shaped grid of wires, as shown in a figure below. Nodes of the grid are at points (X, Y), where X and Y are integers from 0 to N−1. An electric current flows through the grid, between the nodes at (0, 0) and (N−1, N−1).



Initially, all the wires conduct the current, but the wires burn out at a rate of one per second. The burnouts are described by three zero-indexed arrays of integers, A, B and C, each of size M. For each moment T (0 ≤ T < M), in the T-th second the wire between nodes (A[T], B[T]) and:

* (A[T], B[T] + 1), if C[T] = 0 or
* (A[T] + 1, B[T]), if C[T] = 1

burns out. You can assume that the arrays describe existing wires, and that no wire burns out more than once. Your task is to determine when the current stops flowing between the nodes at (0,0) and (N−1,N−1).

Write a function:

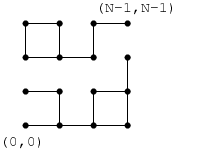
int solution(int N, int A[], int M, int B[], int M2, int C[], int M3);

that, given integer N and arrays A, B and C, returns the number of seconds after which the current stops flowing between the nodes at (0, 0) and (N−1, N−1). If the current keeps flowing even after all M wires burn out, the function should return −1.

For example, given N = 4, M = 9 and the following arrays:

A[0] = 0 B [0] = 0 C[0] = 0 A[1] = 1 B [1] = 1 C[1] = 1 A[2] = 1 B [2] = 1 C[2] = 0 A[3] = 2 B [3] = 1 C[3] = 0 A[4] = 3 B [4] = 2 C[4] = 0 A[5] = 2 B [5] = 2 C[5] = 1 A[6] = 1 B [6] = 3 C[6] = 1 A[7] = 0 B [7] = 1 C[7] = 0 A[8] = 0 B [8] = 0 C[8] = 1

your function should return 8, because just after the eighth wire burns out, there is no connection between the nodes at (0, 0) and (N−1, N−1). This situation is shown in the following figure:



Given N = 4, M = 1 and the following arrays:

A[0] = 0 B [0] = 0 C[0] = 0

your function should return −1, because burning out a single wire cannot break the connection between the nodes at (0, 0) and (N−1, N−1).

Assume that:

* N is an integer within the range [1..400];
* M is an integer within the range [0..2\*N\*(N−1)];
* each element of arrays A, B is an integer within the range [0..N−1];
* each element of array C is an integer that can have one of the following values: 0, 1.

Complexity:

* expected worst-case time complexity is O(N2\*log(N));
* expected worst-case space complexity is O(N2), beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.

Facebook Hackercup — Find the min

After sending smileys, John decided to play with arrays. Did you know that hackers enjoy playing with arrays? John has a zero-based index array, m, which contains n non-negative integers. However, only the first k values of the array are known to him, and he wants to figure out the rest.

John knows the following: for each index i, where k <= i < n, m[i] is the minimum non-negative integer which is **not** contained in the previous **k** values of m.

For example, if k = 3, n = 4 and the known values of m are [2, 3, 0], he can figure out that m[3] = 1.

John is very busy making the world more open and connected, as such, he doesn't have time to figure out the rest of the array. It is your task to help him.

Given the first k values of m, calculate the nth value of this array. (i.e. m[n - 1]).

Because the values of n and k can be very large, we use a pseudo-random number generator to calculate the first k values of m. Given non-negative integers a, b, c and positive integer r, the known values of m can be calculated as follows:

m[0] = a   
m[i] = (b \* m[i - 1] + c) % r, 0 < i < k

**Input**

The first line contains an integer T (T <= 20), the number of test cases. This is followed by T test cases, consisting of 2 lines each. The first line of each test case contains 2 space separated integers, n, k (1 <= k <= 10^5, k < n <= 10^9). The second line of each test case contains 4 space separated integers a, b, c, r (0 <= a, b, c <= 10^9, 1 <= r <= 10^9).

**Output**

For each test case, output a single line containing the case number and the nth element of m.