A Simple Quiz on Hand-Written Code

March 19, 2021

Berland Crosswordtime

limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Berland crossword is a puzzle that is solved on a square grid with n rows and n columns. Initially all the cells are white.
To solve the puzzle one has to color some cells on the border of the grid black in such a way that:

- exactly U cells in the top row are black;
- exactly R cells in the rightmost column are black;
- exactly D cells in the bottom row are black;
- exactly L cells in the leftmost column are black.

Note that you can color zero cells black and leave every cell white.
Your task is to check if there exists a solution to the given puzzle. InputThe first line contains a single integer t ($1 \leq t \leq 1000$) — the number of testcases. Then the descriptions of t testcases follow. The only line of each testcase contains 5 integers n, U, R, D, L ($2 \leq n \leq 100$; $0 \leq U, R, D, L \leq n$). OutputFor each testcase print "YES" if the solution exists and "NO" otherwise. You may print every letter in any case you want (so, for example, the strings yEs, yes, Yes and YES are all recognized as positive answer).
Sample Input:
4
5 2 5 3 1
3 0 0 0 0
4 4 1 4 0
2 1 1 1 1
Sample Output:
YES
YES
NO
YES

Link

(https://codeforces.com/contest/1494/problem/B)

```
public class Main {
    static Scanner in = new Scanner(System.in);
    static PrintWriter out = new PrintWriter(new OutputStreamWriter(System.out)
    public static void main(String[] args) {
        int t = in.nextInt();
        while (t-- > 0) {
            solve();
        }
        out.close();
    }
    static void solve() {
        int n = in.nextInt();
        int U = in.nextInt();
        int R = in.nextInt();
        int D = in.nextInt();
        int L = in.nextInt();
        int u, r, d, l;
        for (int i = 0; i < 16; i++) {
            u = U - ((i >> 0) &1) - ((i >> 3) &1);
            r = R - ((i >> 0) &1) - ((i >> 1) &1);
            d = D - ((i >> 1) &1) - ((i >> 2) &1);
            l = L - ((i >> 2) &1) - ((i >> 3) &1);
            if (
                    u >= 0 && u <= n-2 &&
                    l >= 0 && l <= n-2 &&
                    d >= 0 && d <= n-2 &&
                    r >= 0 && r <= n-2
            ) {out.println("YES"); return;}
        }
        out.println("NO");
    }
}
```