

Recolectando basura

Contribución de Fabricio Loor

Descripción del problema

Ruperto es el encargado de la limpieza en un complejo de oficinas, representado por una grilla de $N \times M$ cuartos. El cuarto de la esquina superior derecha es donde está el basurero y el resto son oficinas, cada una de las cuales está ocupada por un trabajador.

Cada oficina puede o no tener basura para recolectar. Ruperto sólo puede recolectar basura de a una oficina por vez, y para hacerlo debe

- entrar al cuarto donde está el basurero para agarrar una nueva bolsa de basura vacía,
- moverse por los cuartos desde el basurero hasta la oficina donde se va a recolectar la basura, pudiendo en cada paso pasar solamente de un cuarto a otro adyacente,
- llenar la bolsa con la basura recolectada de esa oficina, y
- volver al basurero y descartar la bolsa.

Cada trabajador que está en una oficina con basura para recolectar tiene un cierto valor de enojo, representado con un entero positivo que depende de cuánta basura acumulada hay allí. **Cada vez que uno de estos trabajadores ve a Ruperto entrar con una bolsa de basura vacía a un cuarto adyacente**, su enojo se incrementa en 1, porque se impacienta y desea que Ruperto recolecte su basura inmediatamente.

Debes escribir una función que calcule el mínimo enojo total posible, si Ruperto realiza la recolección con un plan adecuado. El enojo total se define como la suma de todos los enojos máximos de los trabajadores de las oficinas, es decir, su enojo en el momento en el que finalmente se recolectó la basura de su oficina.

Nota: dos cuartos son adyacentes si comparten una pared, es decir si uno está inmediatamente abajo, arriba, a la derecha o a la izquierda del otro en la grilla.

Descripción de la función

Se debe implementar una función `recolector(E)`. Sus parámetros son:

- E : Arreglo de N arreglos de M enteros cada uno, que forman una matriz de $N \times M$ que indica si hay basura para recolectar y los niveles de enojo de los empleados de las oficinas:
 - $E[i][j] = -1$ en el basurero, **que es siempre la esquina superior derecha**,
 - $E[i][j] = 0$ en las oficinas donde no hay basura para recolectar, y
 - $E[i][j] > 0$ en las oficinas donde sí hay basura para recolectar, en cuyo caso $E[i][j]$ es el nivel de enojo inicial.

La función debe retornar un único entero: el mínimo valor posible del enojo total.

 7	 10	 -1
 6	 0	 9
 2	 3	 9
 4	 9	 0

Evaluador local

El evaluador local lee de la entrada estándar con el siguiente formato:

- Dos enteros N, M
- N líneas de M enteros cada una, con los enteros $E[i][j]$ ($0 \leq i < N$, $0 \leq j < M$)

Escribe en la salida estándar el valor retornado por la función `recolector`.

Restricciones

$$1 \leq N, M \leq 1000$$

$$E[0][M-1] = -1$$

$$0 \leq E[i][j] \leq 10 \text{ si } (i, j) \neq (0, M-1)$$

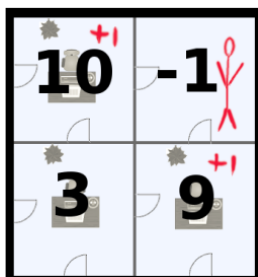
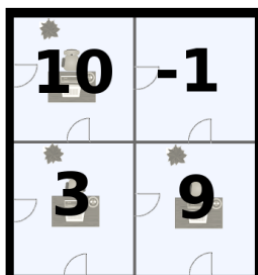
Ejemplos

Si la entrada es:

2	2
10	-1
3	9

28

Se muestra a continuación un ejemplo paso a paso de cómo podría Ruperto conseguir enojo total $28 = 10 + 12 + 3$



Si la entrada es:

4	3
7	10 -1
6	0 9
2	3 9
4	9 0

La salida correcta es:

78

En este ejemplo, el enojo total se obtiene luego de sumar: $10+12+11+9+8+5+11+5+7$, que se puede obtener si por ejemplo se van limpiando las oficinas en el siguiente orden: $(1, 2)$, $(0, 1)$, $(2, 2)$, $(0, 0)$, $(1, 0)$, $(2, 1)$, $(3, 1)$, $(2, 0)$, $(3, 0)$

En la siguiente figura se indica este orden con números en rojo.



Subtareas

1. $NM = 2$ (6 puntos)
2. $N = 1$ (16 puntos)
3. $N \leq 2$ (16 puntos)
4. $N \leq 50, M \leq 50$ (21 puntos)
5. $E[i][j] = 1$ si $(i, j) \neq (0, M - 1)$ (20 puntos)
6. Sin más restricción (21 puntos)