

El turno de Khris

Contribución de Facundo Gutiérrez y Agustín Santiago Gutiérrez

Descripción del problema

Shining Force es un videojuego de fantasía de tipo RPG táctico, lanzado por primera vez en 1992.

Durante el transcurso del juego ocurren batallas entre las fuerzas del mal y un grupo de valientes luchadores liderados por el jugador. Cada luchador posee al comenzar la batalla una cierta cantidad de **puntos de vida**. A medida que los luchadores son golpeados por el enemigo pierden puntos de vida.

El juego se desarrolla por turnos en una cuadrícula de r filas y c columnas: cada luchador ocupa una cierta casilla. Las filas se numeran desde 0 hasta $r - 1$ y las columnas se numeran desde 0 hasta $c - 1$.

Los luchadores se dividen en diferentes clases de acuerdo a sus habilidades: hay espadachines agresivos, poderosas hechiceras, arqueras élficas, enanos robustos con grandes hachas, hombres pájaro y muchas clases más de luchadores.

Una de las clases más importantes en el juego es la clase de los **sanadores**. Los sanadores son luchadores que tienen la capacidad especial de sanarse a sí mismos y a otros luchadores, haciendo que **recuperen puntos de vida** perdidos durante el combate. Entre los mejores sanadores del juego se encuentra Khris.

En este problema, Khris tiene la capacidad de recuperar **hasta un máximo de 10 puntos de vida** de un luchador elegido que esté a **una distancia de a lo sumo 2 casillas** de Khris. La distancia entre las casillas (i_1, j_1) e (i_2, j_2) es $|i_1 - i_2| + |j_1 - j_2|$, que corresponde a la cantidad de movimientos en horizontal y vertical necesarios para pasar de una casilla a la otra.

Nunca está permitido que un luchador alcance más puntos de vida que los que tenía al comenzar la batalla, sin importar cuántas veces sea sanado.

En su turno, Khris puede realizar hasta un máximo de d movimientos entre casillas adyacentes (puede quedarse en la misma

casilla en la que inicia el turno). Dos casillas son adyacentes si comparten un lado, de modo que una casilla tiene como máximo 4 casillas adyacentes.

Al moverse, Khris puede pasar por casillas que contengan otros luchadores, pero no puede pasar por casillas que contengan obstáculos, ni salirse fuera de la cuadrícula. Luego de estos movimientos, Khris **debe quedar ubicada finalmente en una casilla vacía**: no puede quedar en una casilla donde ya haya otro luchador. Desde esta casilla final, para completar el turno, puede utilizar su capacidad de sanar a cualquier luchador que se encuentre a lo sumo a 2 casillas de distancia. Puede incluso sanarse a sí misma.

Debes implementar una función que dadas las posiciones, puntos de vida iniciales y puntos de vida actuales de cada uno de los n luchadores y un mapa con los obstáculos, determine la máxima cantidad posible de puntos de vida que es posible recuperar en este turno de Khris.

Los luchadores se numeran desde 0 hasta $n-1$ y consideramos que Khris es la luchadora número 0.

La siguiente figura ilustra el primer ejemplo. La casilla que contiene a Khris se encuentra indicada con un recuadro blanco.



Detalles de implementación

Se debe implementar una función:

`sanar(fila,col,vidaInicial, vidaActual, d, mapa)`, donde:

- `fila,col`: Arreglos de n enteros, que indican las coordenadas $(fila[i], col[i])$ del i -ésimo luchador.
- `vidaInicial, vidaActual`: Arreglos de n enteros. $vidaInicial[i]$ es la cantidad de puntos de vida iniciales del luchador i , y $vidaActual[i]$ es la cantidad de puntos de vida que tiene actualmente el luchador i .
- `d`: Un entero que indica la cantidad de movimientos que puede hacer Khris en su turno
- `mapa`: Un arreglo de r cadenas de texto, todas de c caracteres, que describen los obstáculos. El caracter j -ésimo de la cadena `mapa[i]` será "X" si (i, j) es una casilla con un obstáculo, y será "." en caso contrario.

La función debe retornar un único entero: la máxima cantidad de puntos de vida que es posible sanar.

Evaluador local

El evaluador local lee de la entrada estándar:

- Una línea con tres enteros n, r y c
- n líneas, cada una con cuatro enteros `fila[i], col[i], vidaInicial[i]` y `vidaActual[i]`
- Una línea con un entero d
- r líneas más. La i -ésima de estas líneas tiene la cadena `mapa[i]`

Escribe el entero retornado por la función.

Restricciones

- $1 \leq r, c, n \leq 1000$
- $1 \leq d \leq 1.000.000$
- $0 \leq fila[i] \leq r - 1$
- $0 \leq col[i] \leq c - 1$
- $1 \leq vidaActual[i] \leq vidaInicial[i] \leq 100$
- `mapa[i][j]` será "X" o "."
- Todas las posiciones $(fila[i], col[i])$ de los luchadores son diferentes entre sí, y son posiciones sin obstáculos en el mapa.

Ejemplos

Si el evaluador local recibe la siguiente entrada:

```
6 4 6
2 2 13 10
3 3 40 34
0 5 40 1
2 0 50 48
0 4 40 32
3 1 6 1
4
..XX..
...XXX
...X..
.....
```

Con una solución correcta escribe:

8

Lo que se obtiene sanando al luchador de la posición (0, 4) desde la posición (2, 4). Si en cambio fuera $d = 5$ en lugar de $d = 4$ en este ejemplo, la respuesta sería 10 porque sería posible sanar al luchador de la posición (0, 5) desde la posición (2, 5).

Si en cambio se recibe:

```
2 2 2
0 0 13 13
1 1 40 40
5
..
..
```

La salida correcta es:

0

Subtareas

1. $n = 1$ (4 pts.)
2. $n = 2$ y no hay obstáculos (6 pts.)
3. $r = 1$ (27 pts.)
4. $r, c \leq 10$ (18 pts.)
5. $n = 2$ (20 pts.)
6. Sin más restricción (25 pts.)