Contest: Marathon Match 67

Normal view

Problem: RockyMine

Problem Statement

You are given a map of a gold mine. Each cell may contain a piece of gold and/or several layers of rocks. You can move through cells without rocks, destroy rocks using dynamite cartridges of several types and collect gold from rocks-free cells. The number of moves you can perform and the number of dynamite cartridges of each type that you can use is limited. Given all these limitations, your task is to get as much gold as you can.

Implementation

The map of the mine is a rectangular section divided in cells, **W** cells wide (east-west direction) and H cells high (north-south direction). Your code should implement a single method **collectGold**(vector <int> dynamite, vector <int> effect, int W, vector <int> gold, vector <int> rocks). Its parameters give you the following information:

- **dynamite** describes the set of dynamite cartridges you have when entering the mine: **dynamite**[k] is the number of dynamite cartridges of type k you have before the first move. You can figure out the number of types of dynamite D as the number of elements in **dynamite**.
- **effect** describes the effect of exploding one unit of dynamite of each type: **effect**[i*5+j+25*k] (for i=0..4, j=0..4, k=0..D-1) is the number of layers of rocks that will be destroyed in cell (row-2+i,col-2+j) after using one dynamite cartridge of type k at cell (row,col). If cell (row-2+i, col-2+j) is outside the map, the effect at this cell is ignored. For example, array {3,2,0,2,3,2,4,1,4,2,0,1,0,1,0,2,4,1,4,2,3,2,0,2,3} describes the damage of one type of cartridges which goes as follows:

```
3 2 0 2 3
2 4 1 4 2
0 1 0 1 0
2 4 1 4 2
3 2 0 2 3
```

- **W** is the width of the mine. You can figure out the height of the mine H as **gold**.length/**W**.
- **gold** and **rocks** describe maps of gold and rocks in the mine: **gold**[row*W+col] is the value of the piece of gold in cell (row,col), and **rock**[row*W+col] is the number of layers of rocks in this cell. Cells with row = 0, H-1 or col = 0, W-1 will contain neither gold nor rocks.
- **maxMoves** is the maximum number of moves you are allowed to make.

You start at cell (0,0) (north-west corner of the map). In one move you can:

- move one cell east (use 'E'), west ('W'), south ('S') or north ('N'). You can move only through cells which have 0 layers of rock. When you move to a cell which contains a piece of gold, you pick it up, the piece is removed from the cell, and its value is added to the total value of gold

you've collected. It's not allowed to move outside the map.

- put one dynamite cartridge at your current position by using digit 'k', where k is 0-based index of dynamite type to use. The cartridge explodes 5 moves after putting it: if you put the cartridge as move i, the rocks are destroyed between moves i+5 and i+6. If immediately after move i+5 you are in a cell where rocks will be destroyed with the explosion (i.e., the corresponding element of **effect** is positive), you get hurt and your score for the test case is 0. If a cell contains n layers of rocks and the explosion destroys at least n+1 layers of rocks, then all rocks in this cell will be destroyed and, if there's a piece of gold in this cell, it will be destroyed as well.
- stay where you are by using '-'.

The return from the method should be the list of at most **maxMoves** moves you'll do while exploring the mine, in order. The i-th character of your return gives your choice of move i. After the exploration is over (all moves of the return are executed), you stay where you are until all dynamite cartridges you've put explode, and than you automatically get out of the mine using any gold-free route (this is always possible), so you can still lose your gold (if you're damaged with an explosion), but can't acquire it.

Scoring

Your score for each test case will be the total value of gold you have collected. Invalid return of any kind (using a type of dynamite you've run out of, using invalid character for move, moving outside of the map or to a cell with non-zero number of layers of rocks, using too many moves etc.) results in 0 score. Your overall score will be the sum of your individual scores, each of them divided by the greatest score achieved by anyone for that test case.

Visualizer

A visualization tool is provided for offline testing. It also allows manual play.

Definition

Class:	RockyMine
Method:	collectGold
Parameters:	vector <int>, vector <int>, int, vector <int>, vector <int>, int
Returns:	string
Method signature:	string collectGold(vector <int> dynamite, vector <int> effect, int W, vector <int> gold, vector <int> rocks, int maxMoves)

(be sure your method is public)

Notes

- The memory limit is 1024 MB and the time limit is 10 seconds per test case (which includes only time spent in your code). There is no explicit code size limit.
- There are 10 example test cases and 100 full submission test cases.

Constraints

- All variables are generated randomly and independently, and have uniform distribution unless otherwise stated.
- **W** and H will be between 10 and 100, inclusive.
- In each cell of the map, **rocks** will be between 0 and 9, inclusive, and **gold** will be generated as a normally distributed Gaussian value, multiplied by 10, rounded to the nearest integer and truncated to fit within [0, 63] interval (negative values are replaced with 0, values larger than 63 - with 63).
- D will be between 2 and 7, inclusive.
- For each type of dynamite, its quantity will be between 1 and (H-2)*(**W**-2)/24+1 (integer division), and its effect will be as a matrix with 0 damage at the center and 0..4 damage at other cells. Additionally, the matrix will be symmetrical with respect to the middle row, the middle column and both primary and secondary diagonals.
- **maxMoves** will be between (**W***H)/8 (integer division) and **W***H.

Examples

0)

```
seed = 1
H = 10
W = 10
maxMoves = 48
D = 6
```

picture

1)

```
seed = 2
H = 50
W = 23
maxMoves = 685
D = 5
```

picture

2)

```
seed = 3
H = 29
W = 99
maxMoves = 1384
D = 2
```

picture

3)

```
seed = 4
H = 83
W = 52
maxMoves = 3121
D = 2
```

picture

4)

```
seed = 5
H = 59
W = 83
maxMoves = 3696
D = 4
```

picture

5)

```
seed = 6
H = 36
W = 34
maxMoves = 519
D = 6
```

picture

6)

```
seed = 7
H = 54
W = 98
maxMoves = 1664
D = 4
```

picture

7)

```
seed = 8
H = 30
W = 49
maxMoves = 676
D = 2
```

picture

8)

```
seed = 9
H = 74
W = 55
maxMoves = 2920
D = 2
```

picture

9)

```
seed = 10
H = 41
W = 84
maxMoves = 2518
D = 3
```

picture
