Pavimento V25712_es

Es necesario pavimentar un recorrido dentro de una matriz. En cada posición de la matriz habrá un valor entero que contendrá el valor actual de alquitrán en dicha posición. Debéis hacer una función que reciba una matriz `M`, una cantidad de `quitra > 0`, un valor `limit > 0` de alquitrán por posición y un string `recorregut` que podrá contener únicamente los caracteres `N`, `S`, `E`, `O`. Estos valores determinan un recorrido por la matriz: partiendo de la posición inicial $(0, 0)$, si recibimos una `N` tendremos que subir una fila, si recibimos una `S` tendremos que bajar una fila, si recibimos una `O` tendremos que ir una columna a la izquierda, y si recibimos una `E` tendremos que ir una columna a la derecha. Obviamente, no podemos salir de la matriz. Si el carácter nos hace salir de la matriz, deberemos quedarnos en la misma posición.
La función deberá colocar `limit` unidades de alquitrán en cada posición del `recorregut`, utilizando, como máximo, el valor de `quitra` del que disponemos. Al final, informará de si hemos podido pavimentar el `recorregut` o no.

La posición de partida es $(0, 0)$. A partir de aquí, la primera posición que visitaremos será la que indique la primera letra del `recorregut`. Cada vez que pasamos por una posición (i, j) puede ocurrir

1. Que contenga un valor de alquitrán superior a `limit`. En este caso, **reducimos** a `limit` el valor de alquitrán de esta posición, y el excedente lo acumulamos en `quitra` (lo podremos usar para otras posiciones).

2. Que contenga un valor de alquitrán inferior a `limit`. En este caso será necesario poner dicha posición a `limit` con el `quitra` que tenemos. Si no tenemos suficiente, pondremos el alquitrán que nos quede.

Podemos pasar por la misma posición más de una vez, dependiendo de lo que nos indique el `recorregut`. Ahora bien, si estamos en una posición determinada y el siguiente movimiento del `recorregut` nos hace salir de la matriz, lo que haremos será no movernos de posición y no hacer nada más.
La función devolverá `PAVIMENTAT` si hemos podido pavimentar todo el `recorregut`. En cambio, si no hemos tenido suficiente alquitrán para poder pavimentar todo el `recorregut`, deberá devolver `NO_PROU_QUITRA`. Además, deberá modificar la matriz `M` con la nueva pavimentación.

Por ejemplo, si tenemos la matriz

1	2	-1	3
-1	1	1	-4
2	4	-1	2
3	-1	-1	2
1	0	1	1

y es necesario pavimentar `recorregut = EESESSSSS` con `quitra = 60` y `limit = 5`, comenzaremos en la posición $(0, 0)$ (que no tocaremos). El recorrido será el siguiente:

Recorregut	Posición	Quitrà
E	(0,1)	57
E	(0,2)	51
S	(1,2)	47
E	(1,3)	38
S	(2,3)	35
S	(3,3)	32
S	(4,3)	28
S	(4,3)	28
S	(4,3)	28

El resultado será `PAVIMENTAT`, y la matriz que quedará es:

1	5	5	3
-1	1	5	5
2	4	-1	5
3	-1	-1	5
1	0	1	5

Haz una función `matriu_paviment` con la siguiente declaración y especificación:

```
/*
 * PRE:  M.size() > 0 and M[0].size() > 0, es una matriz de enteros.
 *  quitra > 0, limit > 0.
 *  recorregut.size() > 0 y recorregut solo contiene 'N','S','E','O'.
 *
 * POST: Devuelve PAVIMENTAT si hemos podido pavimentar todo el recorrido.
 *   Devuelve NO_PROU_QUITRA si no tenemos suficiente alquitrán para pavimentar
 *               el recorrido.
 *   M quedará modificada con todo el recorrido que se haya podido pavimentar.
 */

string matriu_paviment(Matriu& M,int q, int l,
                       const string& recorregut)
```

Observación

Solo es necesario enviar la función que se os pide (y las funciones que hayáis podido declarar vosotros).
Además, en el fichero que enviéis **debe** aparecer también esto:

```
#include <iostream>
#include <vector>
using namespace std;

typedef vector <int>   Vector;
typedef vector<Vector> Matriu;
```

No se puede usar la ordenación de C++: `std::sort`. Tampoco se puede usar el método `push_back()` de la clase `vector`.
Si queréis, podéis usar las funciones `min`, `max` o `swap`.

Entrada

La entrada es una matriz de enteros no vacía, un entero quitrà, un entero límite y un string recorregut. $quitra > 0$, $limit > 0$. $recorregut.size() > 0$ y recorregut solo contiene 'N','S','E','O'.

Salida

Devuelve PAVIMENTAT si hemos podido pavimentar todo el recorrido. Devuelve NO_PROU_QUITRA si no tenemos suficiente alquitrán para pavimentar el recorrido. Modifica la matriz `M` con el nuevo pavimentado.

Ejemplo de entrada 1

```
5 4
 1  2 -1  3
-1  1  1 -4
 2  4 -1  2
 3 -1 -1  2
 1  0  1  1

10
2
OSONOSSSSEEENNO
```

Ejemplo de salida 1

```
   2   2  -1   3
   2   1   1  -4
   2   4   1   2
   2  -1  -1   2
   2   2   2   2
QUITRA: 10 LIMIT: 2 RECORREGUT: OSONOSSSSEEENNO RESULTAT: NO_PROU_QUITRA
```

Ejemplo de entrada 2

```
5 4
 1  2 -1  3
-1  1  1 -4
 2  4 -1  2
 3 -1 -1  2
 1  0  1  1


5
2
OOOOSEEES
```

Ejemplo de salida 2

```
   1   2  -1   3
   2   2   2  -4
   2   4  -1   2
   3  -1  -1   2
   1   0   1   1
QUITRA: 5 LIMIT: 2 RECORREGUT: OOOOSEEES RESULTAT: NO_PROU_QUITRA
```

Ejemplo de entrada 3

```
5 4
 1  2 -1  3
-1  1  1 -4
 2  4 -1  2
 3 -1 -1  2
 1  0  1  1

60
5
EESESSSSS
```

Ejemplo de salida 3

```
   1   5   5   3
  -1   1   5   5
   2   4  -1   5
   3  -1  -1   5
   1   0   1   5
QUITRA: 60 LIMIT: 5 RECORREGUT: EESESSSSS RESULTAT: PAVIMENTAT
```

Ejemplo de entrada 4

```
5 4
0  2  4  8
29 -1 4  -1
28 39 35 15
27 -7 36 -6
25 22 20 20

90
10

SEOOSSSOEESEEENNN
```

Ejemplo de salida 4

```
   0   2   4   8
  10  10   4  10
  10  39  35  10
  10  -7  36  10
  10  10  10  10
QUITRA: 90 LIMIT: 10 RECORREGUT: SEOOSSSOEESEEENNN RESULTAT: PAVIMENTAT
```

Información del problema

Autoría: PRO1

Generación: 2026-01-25T13:06:56.742Z