
Paviment**V25712_ca**

Cal enquitranar un recorregut dins d'una matriu. A cada posició de la matriu hi haurà un valor enter que contindrà el valor actual de quitrà en aquella posició. Heu de fer una funció que rebi una matriu `M`, una quantitat de quitrà `> 0`, un valor `limit > 0` de quitrà per posició i un string `recorregut` que podrà contenir només els caràcters `N`, `S`, `E`, `O`. Aquests valors determinen un recorregut per la matriu: partint de la posició inicial `(0,0)`, si rebem una `N` haurem de pujar una fila, si rebem una `S` haurem de baixar una fila, si rebem una `O` haurem d'anar una columna a l'esquerra, i si rebem una `E` haurem d'anar una columna a la dreta. Òbviament, no podem sortir de la matriu. Si el caràcter ens fa sortir de la matriu, ens haurem de quedar a la mateixa posició.

La funció haurà de posar `limit` unitats de quitrà a cada posició del `recorregut`, fent servir, com a màxim, el valor de `quitrà` que tindrem. Al final, informarà de si hem pogut enquitranar el `recorregut` o no.

La posició de partida és `(0,0)`. A partir d'aquí, la primera posició que visitarem serà la que indiqui la primera lletra del `recorregut`. Cada vegada que passem per una posició `(i,j)` podrà passar

1. Que contingui un valor de quitrà superior a `limit`. En aquest cas, **reduïm** a `limit` el valor de quitrà d'aquesta posició, i l'excedent l'acumulem a `quitrà` (el podrem fer servir per a altres posicions).
2. Que contingui un valor de quitrà inferior a `limit`. En aquest cas caldrà posar aquesta posició a `limit` amb el `quitrà` que tenim. Si no en tenim prou, hi posem el quitrà que ens queda.

Podem passar per la mateixa posició més d'una vegada, depenent del que ens mani fer el `recorregut`. Ara bé, si som en una posició determinada i el següent moviment del `recorregut` ens fa anar fora de la matriu, el que farem serà no moure'ns de posició i no fer res més.

La funció tornarà `PAVIMENTAT` si hem pogut pavimentar tot el `recorregut`. En canvi, si no hem tingut prou quitrà per a poder pavimentar tot el `recorregut`, haurà de tornar `NO_PROU_QUITRA`. A més, haurà de modificar la matriu `M` amb la nova pavimentació.

Per exemple, si tenim la matriu

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

i cal pavimentar `recorregut = ESESSSSS` amb `quitrà = 60` i `limit = 5`, començarem a la posició `(0,0)` (que no tocarem). El recorregut serà aquest:

Recorregut	Posició	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 resultat serà PAVIMENTAT, i la matriu que quedarà és:

```

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

```

Fes una funció `matriu_paviment` amb la següent declaració i especificació:

```

/*
 * PRE:  M.size() > 0 and M[0].size() > 0, és una matriu d'enters.
 *  quitra > 0, limit > 0.
 *  recorregut.size() > 0 i recorregut només conté 'N','S','E','O'.
 *
 * POST: Retorna PAVIMENTAT hem pogut pavimentar tot el recorregut.
 *  Retorna NO_PROU_QUITRA si no tenim prou quitrà per a pavimentar
 *           el recorregut.
 *  M estarà modificada amb tot el recorregut que s'hagi pogut pavimentar.
 */

```

```

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

```

Observació

Només cal enviar la funció que us demanem (i les funcions que hagueu pogut declarat vosaltres).

A més, al fitxer que envieu **cal** que també hi hagi això:

```

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

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

```

No es pot fer servir l'ordenació del C++: `std::sort`. Tampoc no es pot fer servir el mètode `push_back()` de la classe `vector`.

Si voleu, podeu fer servir les funcions `min`, `max` o `swap`.

Entrada

L'entrada és una matriu d'enters no buida, un enter quitrà, un enter límit i un string recorregut. $quitrà > 0$, $límit > 0$. $recorregut.size() > 0$ i recorregut només conté 'N','S','E','O'.

Sortida

Retorna PAVIMENTAT hem pogut pavimentar tot el recorregut. Retorna NO_PROU_QUITRA si no tenim prou quitrà per a pavimentar el recorregut. Modifica la matriu M amb el nou pavimentat.

Exemple d'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
```

Exemple de sortida 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
```

Exemple d'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
OOOOSEEEES
```

Exemple de sortida 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: OOOOSEEEES RESULTAT: NO_PROU_QUITRA
```

Exemple d'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
```

Exemple de sortida 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
```

Exemple d'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

SEOSSSOEESEENNN
```

Exemple de sortida 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: SEOSSSOEESEENNN RESULTAT: PAVIMENTAT
```

Informació del problema

Autoria: PRO1

Generació: 2026-01-25T13:06:52.426Z

© *Jutge.org*, 2006–2026.
<https://jutge.org>