Simulant el chip-firing game — X31871_ca

GRAU-PRO1, FIB (2014-12)

El *chip-firing game* sobre matrius és un joc matemàtic que modela el balancejat de càrrega en una estructura bidimensional. Donada una matriu quadrada amb n files i n columnes, cada posició d'aquesta emmagatzema inicialment un cert nombre de chips i una llista de les posicions veïnes accessibles amb les quals està connectada i interacciona.

El joc considera el repartiment dels chips al llarg del temps sobre les posicions de la matriu. La distribució es fa seguint la següent regla local a cadascuna de les posicions: A cada pas de temps, cada posició que tingui com a mínim tants chips com posicions veïnes accessibles, passa un chip a cadascuna d'aquestes posicions.

Definició 1: Per tal de simplificar el joc, considerarem que només les *posicions interiors* de la matriu participen al joc. Donada una matriu quadrada de $n \times n$, es consideren posicions interiors les posicions (i, j) tals que $0 < i < n - 1$ i $0 < j < n - 1$ (és a dir, les posicions que no ocupen ni la primera ni la última fila, ni la primera ni la última columna).

Definició 2: Les posicions veïnes a una posició (i, j) de la matriu són les posicions que es troben a distància 1 d'aquesta, ja sigui horitzontalment, verticalment o bé en diagonal. Observeu que tota posició (i, j) té, com a molt, 8 posicions veïnes. Anomenarem *posicions veïnes accessibles* al subconjunt de posicions veïnes amb el qual hi ha interacció. Tota posició veïna es pot codificar amb dos caràcters d'acord amb l'esquema següent:

NW	NN	NE
WW		EE
SW	SS	SE

Per exemple, la cadena "NNSS" (nord-nord, sud-sud) indicaria que les dues úniques posicions accessibles són la immediatament superior i la immediatament inferior.

Es demana:
Volem fer una simulació seqüencial de l'evolució del chip-firing game en què a cada instant de temps només s'actualitza una posició. L'ordre d'actualització de les posicions es defineix per un recorregut per columnes de la matriu. La simulació només implica posicions interiors i comença a la posició interior de dalt a l'esquerra (la posició $(1, 1)$). La simulació continua fent un recorregut complet per columnes de totes les posicions de la matriu. Anomenarem *volta* al recorregut complet de la matriu.

Feu un programa que, donada una matriu del joc, determini quina serà la matriu del joc després de simular una volta, i indiqui si s'ha produït o no algun canvi a la matriu del joc. A més ha de proporcionar el nombre de chips que s'han traspassat de posicions interiors de la matriu cap a posicions no interiors durant la volta simulada.

El vostre programa ha de representar la matriu de joc mitjançant el següent tipus:

```
struct Cell {
    int contingut;          // nombre de chips
    string veins;           // descripció de les posicions veines accessibles
};

typedef vector< vector<Cell> > Joc;
```

Entrada

L'entrada està formada por un únic cas, composat per un nombre natural $n \geq 3$ que defineix la dimensió de la matriu quadrada. Després s'hi troben n^2 naturals que representen el nombre de chips a cadascuna de les posicions de la matriu. Seguidament es descriuen, per cada posició interior, les posicions veïnes accessibles. Les posicions veïnes accessibles es descriuen mitjançant un string tal com s'indica a la Definició 2.

Sortida

Escriviu el nombre de chips a cada posició de la matriu després de simular-ne una volta i indiqueu si s'ha produït o no algún canvi respecte a la matriu del joc inicial. Indiqueu també el nombre de chips que han estat traslladats de la part interior de la matriu cap a les files i columnes no interiors. Seguiu el format especificat als exemples.

Observació

Noteu que, tot i que només les posicions interiors de la matriu participen a la simulació del joc, les posicions no interiors també poden emmagatzemar chips.

El vostre codi ha de seguir bones normes d'estil i contenir els comentaris que considereu oportuns. Es valorarà la claredat del disseny i la bona utilització de procediments així com l'estil de programació.

Exemple d'entrada 1

```
4
0 0 0 0
1 1 1 1
0 1 1 0
0 0 0 0
SS SS
SS SS
```

Exemple de sortida 1

```
0 0 0 0
1 0 0 1
0 1 1 0
0 1 1 0
canvis: si
chips nous fora: 2
```

Exemple d'entrada 2

```
4
0 0 0 0
1 2 2 1
1 1 1 1
0 0 0 0
SS SS
SS SS
```

Exemple de sortida 2

```
0 0 0 0
1 1 1 1
1 1 1 1
0 1 1 0
canvis: si
chips nous fora: 2
```

Exemple d'entrada 3

```
4
0 0 0 0
1 2 1 1
1 1 1 1
0 0 0 0
SSEENN SSSE
SSEE SSEESW
```

Exemple de sortida 3

```
0 0 0 0
1 2 1 1
1 1 1 1
0 0 0 0
canvis: no
chips nous fora: 0
```

Informació del problema

Autor : Maria J. Serna
Generació : 2014-12-16 14:52:11