

Méthodes de Conception d'Algorithmes

C. TRABELSI & E. MARTINS & M. FRANÇOIS

TP1 -- Virus récursif

Avant propos

Nous allons maintenant utiliser le paradigme de la récursivité sur un autre problème : la propagation de virus. Vous verrez que ce problème, comme celui de la cascade des zéros du démineur, est lui aussi intrinsèquement récursif.

1 Problème : Propagation de Virus

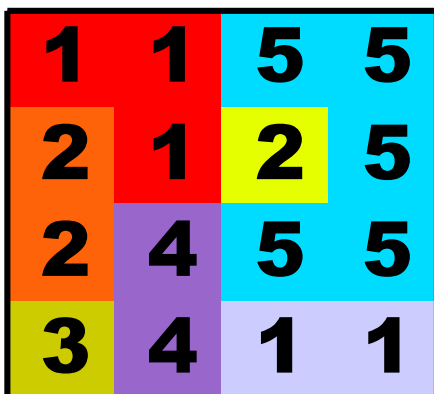
Considérons un tableau bidimensionnel d'entiers représentant une foule de gens par leur niveau de défense immunitaire. Un scientifique fou laisse échapper une souche d'un virus très contagieux au milieu de cette foule. Vous devez y simuler la propagation du virus. Notez que le virus sera capable de se propager par contact¹, avec des personnes dont le niveau immunitaire est égal ou inférieur.

1.1 Boîte à outils : quelques notions utiles

Voici quelques notions pour vous aider.

Connexité : *la connexité est une notion de topologie qui formalise le concept d'« objet d'un seul tenant » : deux cases sont dites connexes si elles se touchent par au moins un côté, une zone est dite connexe si il est possible de se rendre d'une partie de la zone à une autre sans la quitter.*

Composantes connexes : *Une composante connexe est la plus grande zone connexe présentant une caractéristique homogène.*



Sur le schéma ci-contre, chaque composante connexe est délimitée par une zone de couleur différente.

Deux zones de même valeur, qui ne se touchent pas définissent deux composantes connexes différentes.

1. Nous considérerons les 8 voisins d'une case comme étant en contact avec la-dite case.

1.2 Travail à faire

Écrire le programme correspondant aux consignes qui suivent.

Votre programme devra lire en entrée² les caractéristiques de la foule. Ces dernières ainsi que les coordonnées du foyer de départ du virus et sa force seront décrites dans un fichier ayant la forme suivante :

```
5 5
4 4 1 3 5
4 4 5 1 1
3 3 3 4 3
5 1 3 4 5
5 3 1 4 5
2 4
3
```

Les deux premiers nombres indiquent la taille de la foule (hauteur et largeur), les nombres suivants correspondent aux niveaux de défense immunitaire de chaque personne dans la foule. Les deux nombres qui suivent représentent les coordonnées (x, y) du foyer de départ du virus et le dernier nombre représente la force du virus (virulence).

Remarque :

La virulence reste **constante** tout au long de l'infection. Nous cherchons les composantes connexes de toutes les cases voisines de la première, dont la valeur est inférieure ou égale à la virulence.

Votre programme devra écrire en sortie³ la matrice de propagation correspondante (le foyer de départ sera noté par -2 et les personnes contaminées par -1).

Globalement le programme devra contenir :

- une fonction permettant de charger les défenses immunitaires dans un tableau 2D ;
- une fonction qui initialise à 0, la matrice de propagation ;
- une fonction qui affiche un tableau 2D ;
- une fonction récursive permettant la propagation du virus dans la foule ;
- et la fonction `main` du programme.

2. Rappel : lecture sur le flux d'entrée standard avec la fonction `scanf` et redirection du flux à partir d'un fichier (`./EXEC < fichier_entree.txt`).

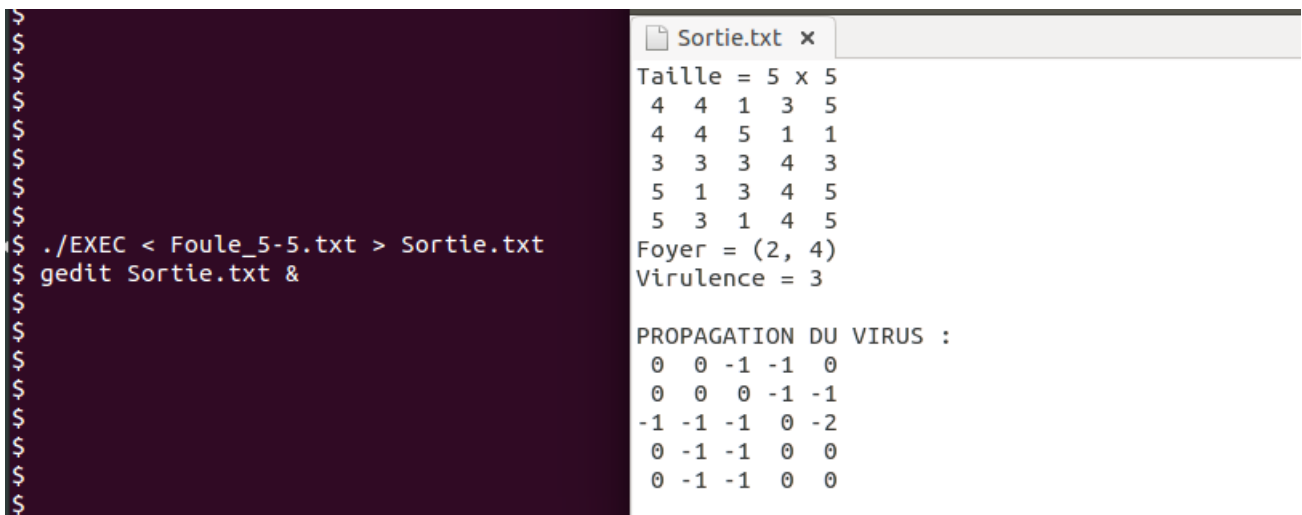
3. Rappel : écriture sur le flux de sortie standard avec la fonction `printf` et redirection du flux vers un fichier (`./EXEC > fichier_sortie.txt`)

Voyons un exemple d'exécution sur le fichier précédent avec une sortie sur la console :

```
$ gcc -o EXEC TP1.c -Wall
$ ./EXEC < Foule_5-5.txt
Taille = 5 x 5
4 4 1 3 5
4 4 5 1 1
3 3 3 4 3
5 1 3 4 5
5 3 1 4 5
Foyer = (2, 4)
Virulence = 3

PROPAGATION DU VIRUS :
0 0 -1 -1 0
0 0 0 -1 -1
-1 -1 -1 0 -2
0 -1 -1 0 0
0 -1 -1 0 0
$
```

Cette fois-ci avec une sortie redirigée vers le fichier `Sortie.txt` :



The image shows a terminal window on the left and a text editor window titled "Sortie.txt" on the right. The terminal window shows the command `./EXEC < Foule_5-5.txt > Sortie.txt` being executed, followed by `gedit Sortie.txt &`. The text editor window displays the output of the program, which is the same as the console output shown in the previous image.

```
$ ./EXEC < Foule_5-5.txt > Sortie.txt
$ gedit Sortie.txt &

Taille = 5 x 5
4 4 1 3 5
4 4 5 1 1
3 3 3 4 3
5 1 3 4 5
5 3 1 4 5
Foyer = (2, 4)
Virulence = 3

PROPAGATION DU VIRUS :
0 0 -1 -1 0
0 0 0 -1 -1
-1 -1 -1 0 -2
0 -1 -1 0 0
0 -1 -1 0 0
```

NB : "Affichage d'un nombre sur 2 caractères"

Pour afficher un nombre sur N caractères quel que soit le nombre de chiffres ou le signe d'un nombre, on utilise le descripteur `"%Nd"`. Par exemple pour 2 caractères : `%2d`.

2 BONUS

- 1. Pour plus de lisibilité, écrire une autre fonction d'affichage permettant d'encadrer le résultat tout en ignorant les personnes non contaminées.

Voilà quelques exemples sur des fichiers plus grands :

```

$ gcc -o EXEC TP1.c -Wall
$ ./EXEC < Foule_15-15.txt
Taille = 15 x 15
6 6 1 8 6 1 2 4 5 2 5 9 9 6 1
7 3 5 2 5 8 7 4 9 5 9 1 2 2 2
4 3 8 8 8 5 2 4 6 6 2 2 7 7 7
7 6 1 5 5 5 5 8 8 8 3 3 3 9 9
6 6 1 1 6 5 5 2 2 9 9 5 2 9 9
6 4 8 1 1 1 9 9 7 7 2 2 9 2 9
8 6 1 7 1 6 6 5 8 6 2 4 1 6 9
1 1 1 5 1 1 1 9 9 5 5 5 8 8 8
2 2 9 9 9 9 5 2 5 4 1 3 8 7 5
1 6 1 2 2 5 8 5 4 1 9 6 3 5 5
8 5 2 7 3 4 5 7 7 5 5 5 8 8 1
4 7 4 8 8 1 6 1 5 6 8 7 7 7 7
2 5 8 4 1 1 8 4 5 5 7 2 9 4 7
9 5 9 5 5 2 5 8 7 4 7 2 9 4 7
9 5 9 4 8 5 2 5 4 2 1 5 7 7 7
Foyer = (0, 14)
Virulence = 5

```

PROPAGATION DU VIRUS :

```

-----
|      -1      -1-1-1-1-1-1      -2|
|  -1-1-1-1      -1  -1  -1-1-1-1|
|-1-1      -1-1-1      -1-1      |
|      -1-1-1-1-1      -1-1-1      |
|      -1-1  -1-1-1-1      -1-1      |
|  -1  -1-1-1      -1-1  -1      |
|      -1  -1      -1      -1-1-1      |
|-1-1-1-1-1-1-1-1      -1-1-1      |
|-1-1      -1-1-1-1-1-1-1      -1|
|-1  -1-1-1-1  -1-1-1      -1-1-1|
|      -1-1  -1-1-1      -1-1-1      -1|
|-1  -1      -1  -1-1      |
|-1-1  -1-1-1  -1-1-1  -1      |
|      -1  -1-1-1-1      -1  -1      |
|      -1  -1  -1-1-1-1-1-1-1      |
-----

```

```

$ ./EXEC < Foule_20-20.txt
Taille = 20 x 20
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
9 9 9 7 7 4 4 4 4 4 4 4 4 4 4 4 4 9 9 9
9 9 9 6 2 5 1 4 2 1 1 1 1 4 4 4 4 9 9 9
9 9 9 3 1 4 2 9 9 9 9 9 9 9 9 9 9 9 9 9
9 9 9 1 2 2 9 9 9 9 9 9 9 9 9 9 9 9 9 9
9 9 9 1 2 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
9 9 9 1 2 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
9 9 9 2 2 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
9 9 9 2 3 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
9 9 9 3 3 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
9 9 9 2 3 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
9 9 9 2 3 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
9 9 9 2 3 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
9 9 9 1 5 3 9 9 9 9 9 9 9 9 9 9 9 9 9 9
9 9 9 2 2 2 3 9 9 9 9 9 9 9 9 9 9 9 9 9
9 9 9 8 2 5 1 4 2 1 1 1 1 4 4 4 4 9 9 9
9 9 9 8 7 5 1 4 2 1 1 1 1 4 4 4 4 9 9 9
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9

Foyer = (2, 16)
Virulence = 5

PROPAGATION DU VIRUS :
-----
|
|      -1-1-1-1-1-1-1-1-1-1-1-2
|      -1-1-1-1-1-1-1-1-1-1-1-1
|      -1-1-1-1
|      -1-1-1
|      -1-1
|      -1-1
|      -1-1
|      -1-1
|      -1-1
|      -1-1
|      -1-1
|      -1-1-1
|      -1-1-1-1
|      -1-1-1-1-1-1-1-1-1-1-1-1
|      -1-1-1-1-1-1-1-1-1-1-1-1
|
|-----

```

- 2. Modifier votre fonction de propagation du virus afin d'observer au fur et à mesure l'évolution de la contamination dans la foule. Pour cela, il suffit de faire appel à la fonction d'affichage et mettre un petit timing. Pour une meilleure visualisation, vous pouvez effectuer un effacement d'écran avant l'appel de la fonction d'affichage.