

Structures de Données et Algorithmes 1
Mini-Projet
Année 2015 – 2016

Étiquetage en Composantes Connexes

N.B. Le projet est réalisé en binôme. Après avoir terminé votre projet, nommez votre répertoire de projet par vos noms, compressez-le avec `tar` ou `zip` et déposez-le sur moodle. Par exemple : `Fuchs_Phan.zip`. Le fichier compressé contiendra :

- les codes sources de votre programme en C. (uniquement les fichiers `.c`, `.h`, pas d'`exe`, ni de `.o`)
- un répertoire de documentation créé par Doxygen
- un fichier `Makefile`

Ce projet est à rendre au plus tard pour **le 8 mai 2016 à 23h00**.

I. Introduction

Une image binaire est une image pour laquelle la valeur de chaque pixel est égale 0 (arrière-plan) ou 1 (premier-plan). On définit dans ce projet un *pixel connecté* comme un pixel du premier-plan (valeur 1) qui a au moins un pixel adjacent (en haute, en bas, à gauche ou à droite) dont la valeur est aussi égale 1.

0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	1	0
0	1	0	0	0	1	0	1	0
0	1	0	0	0	0	1	0	0
0	1	1	0	1	0	1	0	0
0	0	0	0	0	0	0	0	0

Image binaire.

Un ensemble des pixels connectés dans lequel ces pixels sont tous connectés l'un à l'autre est appelé une *composante connexe*. En particulier, un pixel en premier-plan qui ne connecte pas avec tous ses adjacents est aussi une composante connexe. L'objet de ce projet est de faire un étiquetage des composantes connexes.

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

Image contenant les composantes connexes étiquetées par les entiers de 1 à 5.

L'étiquetage en composantes connexes peut être réalisé à l'aide d'une `Pile`. L'algorithme est résumé par les étapes suivantes :

1. Commencez au premier pixel dans l'image, initialisez `etqc = 1` (`etqc` : étiquette courante), initialisez une `Pile`. Allez à l'étape (2).
2. Si ce pixel est en premier-plan et il n'est pas encore étiqueté, puis il sera assigné par la valeur de `etqc`. Ensuite, empilez ce pixel dans la `Pile` et allez à l'étape (3).
Si ce pixel est en arrière-plan, passez au pixel suivant dans l'image et répétez l'étape (2).
3. Dépilez un pixel de la `Pile`, vérifiez les pixels adjacents (en haute, en bas, à gauche et à droite), si un pixel adjacent est en premier-plan et il n'est pas encore étiqueté, assignez-le par la valeur de `etqc` et empilez-le dans la `Pile`. Répétez l'étape (3) jusqu'à il n'y a plus de pixels dans la `Pile`, puis passez à l'étape (4).
4. Augmentez `etqc` par un et revenez à l'étape (2) pour le pixel suivant dans l'image.

(https://en.wikipedia.org/wiki/Connected-component_labeling)

II. Piles

On souhaite d'abord implanter une structure de `Pile` non bornée dans laquelle chaque élément est une paire de nombres naturels $\{x, y\}$ contenant les coordonnées d'un pixel. Créez un fichier `pile.h` pour définir la structure `Pile` et les fonctions de base :

- `pilenouv` : initialisation d'une pile vide
- `empiler` : empilement d'un élément
- `depiler` : dépilement du sommet
- `sommet` : élément du sommet
- `hauteur` : nombre d'éléments
- `vide` : test de vacuité

Implémentez ces fonctions dans un fichier `pile.c`.

III. Étiquetage en composantes connexes

On représente maintenant l'image binaire par une matrice booléenne contenant deux valeurs 0 (arrière-plan) et 1 (premier-plan). On définit aussi une matrice de même taille à la matrice booléenne pour étiqueter les composantes connexes de l'image. Créez deux fichiers `Ecc.h`, `Ecc.c` et réalisez les tâches suivantes :

- 1) Définissez une structure `StrEcc` contenant :
 - une matrice booléenne `matbool`
 - une matrice d'entiers `matecc`
 - deux entiers `m` et `n` qui spécifie la taille de deux matrices
- 2) Écrivez une fonction qui prend deux paramètres `m`, `n` et qui retourne un objet de type `StrEcc` dans lequel les matrices `matbool` et `matecc` de `m` lignes et `n` colonnes sont créées et toutes les valeurs sont initialisées par 0.
- 3) Écrivez une fonction qui prend deux paramètres `k`, `h` et qui retourne `k` entiers aléatoires entre 0 et `h-1`.
- 4) Un entier `k` entre 0 et `m*n-1` peut être transformé à une indice (i, j) d'une matrice de `m` lignes et `n` colonnes. Écrivez une fonction qui prend trois paramètres `k`, `m`, `n` et qui transforme `k` à l'indice (i, j) correspondante.
- 5) Écrivez une fonction permettant d'assigner les valeurs 1 pour `k` éléments aléatoires dans `matbool` à l'aide

de deux fonctions en 3) et 4).

6) Proposez une fonction permettant d'étiqueter les composantes connexes. Le résultat est stocké dans `matecc`.

7) Testez l'algorithme dans un fichier `main.c`

IV. Makefile et Doxygen

Créez un fichier `Makefile` permettant de compiler votre programme. Documentez tous les codes à l'aide de Doxygen. Indiquez l'auteur de chaque fonction.

– fin –