

Initiation à Python

LP ESSIG

2012 – 2013

1 Compter le nombre d’objets d’une image

Le but de cet exercice est de dénombrer le nombre d’objets (ou plutôt de tâches) dans une image. Pour simplifier les choses (et parce que, dans le cas contraire, cela nécessiterait de plus grandes connaissances en traitement d’images) les images seront uniquement binaires. Pour résoudre l’exercice, il suffit donc de compter le nombre de « tâches de 1 » dans une « mare de 0 ».

```
00000000000000000000
00111000000000000000
00011100000011101000
01111100000111011100
01100000000011111100
11000000000111111110
00001111000111111100
00111111100011111100
00000111000000110000
00000000000000000000
```

FIGURE 1 – Exemple d’image binaire à trois objets

2 Un objet, définition

Tout ensemble de pixels ayant la valeur 1 pour lesquels il est possible de tracer un chemin entre chaque paire de pixels est considéré comme un objet. Concrètement, en partant d’un pixel à 1, tous les pixels à 1 qui le touchent et tous ceux qui touchent ceux qui le touchent et ainsi de suite forment un objet.

On considère que deux pixels se touchent s’ils sont adjacent sur une ligne ou une colonne. Les diagonales ne sont pas prises en compte pour déterminer les voisins. Ainsi l’image 2 possède deux objets.

```
000000
110000
111000
110110
100111
001111
```

FIGURE 2 – Illustration du concept de voisinage

3 Travail à réaliser

Écrire un programme qui récupère une image binaire et compte le nombre d'objets qu'elle comporte. Le choix de la méthode d'entrée est laissé à votre appréciation : vous pouvez mettre l'image directement dans votre script, la demander à l'utilisateur avec `input()`, la récupérer depuis la ligne de commande ou la lire dans un fichier avec `open()`. Le morceau de code suivant stocke le contenu du fichier `img.txt` dans la variable `image` :

```
image=''
with open('img.txt') as fichier:
    image=fichier.read()
```

Il est possible de découper une chaîne de caractères en une liste contenant chacune des lignes grâce à : `ligne_par_ligne = image.split()`

4 Mise en œuvre

Pour compter les objets, une technique consiste à enlever les objets un par un tout en les comptant. Pour cela, il faut être capable de délimiter chaque objet.

4.1 Déterminer les pixels intéressants

Une première étape, après avoir récupéré la chaîne de caractères composant l'image, consiste à la parcourir pour déterminer les pixels ayant la valeur 1. À l'issue de cette étape, vous devriez obtenir une liste contenant les couples de coordonnées pour les pixels de valeur 1.

4.2 Trouver un objet et l'éliminer

Pour faciliter le travail de comptage, écrivez une fonction qui élimine un objet de la liste. Cette fonction prend la liste de coordonnées et une coordonnée particulière en paramètre et procède comme suit :

- Si cette coordonnée n'est pas dans la liste, on ne fait rien. Il s'agit du cas où l'on vient de trouver la frontière de l'objet.
- Si elle est dans la liste, on commence par la supprimer, puis on appelle de nouveau la fonction avec les 4 coordonnées voisines.

Au bout d'un moment, toutes les coordonnées voisines auront été éliminées et donc l'objet aussi, par la même occasion.

4.3 Compter les objets

Utilisez la liste de coordonnées et la fonction des questions précédentes pour compter le nombre d'objets dans l'image.

5 Exemple d'images

```

00000000001110000000
11100000001001111000
00111000001111000111
00001110000001111100
00000011100000000111
00000000111000000000
01111000001110000000
01001000000011100000
01111000000000111000
00000000000000000000

```

FIGURE 3 – Exemple d’image binaire à trois objets

```

00000000000000000000
11100000000000000000
10010010010000000000
11110010010000000000
10000011011101110111
10101010010101010101
10111011010101110101
00001000000000000000
00011000000000000000
00000000000000000000

```

FIGURE 4 – Exemple d’image binaire à six objets

```

00000000000000000000
01111111110111111110
01111111100001111110
01111110000000001110
01110001111111110010
01000001111111110000
00001100000000000110
00000111111111111100
00000000111111000000
00000000000000000000

```

FIGURE 5 – Exemple d’image binaire à quatre objets