Esiee-Paris - Algorithmique (IGI-2102) - Projet
R. Natowicz - Mai - Juin 2020

Première partie : coccinelle et pucerons

Sur chaque case d'une certaine grille à L lignes et C colonnes il y a, devinez quoi ? Des petits pucerons, en quantités variées, de ceux dont raffolent les coccinelles. Et qui voit-on sur l'une des cases de la ligne $L-1$? Ça alors ! Une coccinelle, en train de manger le dernier puceron de cette case. Et à y regarder de plus près, on voit que certaines des cases de la grille n'ont plus de pucerons. Elle les a mangés !

Un reporter de Radio-Coccinelle (RC) s'est rendu sur place pour interroger la coccinelle. Elle a déclaré être arrivée sur la grille en volant ; avoir atterri sur une case de la ligne $l = 0$; avoir mangé tous les pucerons qu'elle y avait trouvés ; puis s'être déplacée de case en case vers le Nord, le Nord-Ouest ou le Nord-Est, en prenant bien soin de manger tous les pucerons sur son passage.

Pour conclure elle a déclaré : "J'ai mangé autant de pucerons qu'il était possible d'en manger en montant vers le haut de la grille. Quel festin !"

Le reporter de RC a interviewé la coccinelle sur une case de la ligne $l = L - 1$. Pour que son reportage soit complet, il souhaiterait faire partager à ses auditeurs le périple de la coccinelle depuis son atterrissage sur la grille. Il s'est adressé au Ladybird Observation Satellite (LOS), qui lui a envoyé la photo satellite de la grille, prise quelques secondes avant l'atterrissage de la coccinelle. Les numéros portés sur la grille sont les nombres de pucerons qui étaient sur ses cases.

Voici la grille en question, avant l'atterrissage de la coccinelle sur l'une des cases de la ligne $l = 0$.

7	3	1	2	4	5
6	1	72	3	6	6
5	89	27	10	12	3
4	46	2	8	7	15
3	36	34	1	13	30
2	2	4	11	26	66
1	1	10	15	1	2
$l = 0$	2	4	3	9	6
	$c = 0$	1	2	3	4

Table 1: Nombres de pucerons présents sur chaque case de la grille avant l'arrivée de la coccinelle par la voie des airs.

Questions :

1. Combien de pucerons la coccinelle a-t-elle mangé lors de son périple sur la grille ?
2. Quel a été ce périple ?
3. Écrire un programme `Coccinelle.java` qui :

   1) calcule un tableau $M[0:L][0:C]$ de terme général $M[l][c] = m(l,c)$, nombre maximum de pucerons que la coccinelle peut manger sur un chemin allant d'une case de la ligne $l = 0$ jusqu'à la case (l,c) ;

   2) affiche le chemin de la coccinelle.

Réponse aux questions 1 et 2 ⟶

```
$ javac Coccinelle.java
$ java Coccinelle
Grille des pucerons :
 3  1  2  4  5
 1 72  3  6  6
89 27 10 12  3
46  2  8  7 15
36 34  1 13 30
 2  4 11 26 66
 1 10 15  1  2
 2  4  3  9  6

Tableau M[L][C] de terme général M[l][c] = m(l,c) :
279 277 278 149 145
205 276 145 140 140
204 142 124 134 125
115  71  98 114 122
 64  69  51  90 107
 16  28  35  50  77
  5  14  24  10  11
  2   4   3   9   6

La coccinelle a mangé 279 pucerons.
Elle a suivi le chemin (0,3)(1,2)(2,2)(3,1)(4,0)(5,0)(6,1)(7,0)
Case d'atterrisage = (0,3).
Case de l'interview = (7,0).
$
```

Figure 1: compilation et exécution d'un programme Coccinelle.java de calcul et d'affichage d'un chemin allant de la ligne $l = 0$ à la ligne $l = L - 1$. Ce chemin maximise le nombre de pucerons mangés par la coccinelle.

Deuxième partie : redimensionnement d'images

Le redimensionnement d'images est une opération de réduction ou d'augmentation de la taille d'une image. C'est l'opération qui est à l'oeuvre quand vous modifiez sur votre tablette ou sur votre ordinateur la taille d'une fenêtre contenant des images : les tailles des images sont automatiquement modifiées en conséquence.

En 2007, S. Avidan et A. Shamir publiaient une nouvelle méthode de redimensionnement d'images, "*Seam carving for content-aware image resizing*, ACM Trans.Graph.26,3(2 7),10 – qui devirendra très rapidement la méthode standard.

Cette méthode, *Seam Carving*, est très bien décrite dans l'article mentionné ci-dessus, et dans leur article de 2009 "*Seam carving for media retargeting*" publié dans la revue Communications of the ACM (Association for Computing Machinery) : `https://dl.acm.org/doi/abs/10.1145/1435417.1435437`

La seconde partie du projet est l'implémentation de cette méthode de redimensionnement d'image en langage Java. On ne demande d'implémenter que la réduction de taille.

Votre programme prendra en entrée le nom d'une image au format PNG et les pourcentages entiers de réduction horizontale et verticale.
Exemple : L'exécution de la commande `$ Java SeamCarving chaton.png 20 50` sous Unix ou Linux se traduira par l'écriture d'un fichier `chaton_rezised_20_50.png` dont les nombres de lignes horizontales et verticales de l'image auront été réduits de 20 et 50%.

Votre rapport au format PDF contiendra la description de la méthode, en particulier la fonction que vous avez choisie de minimiser ; les points-clés de votre implémentation ; la commande de compilation de votre programme ; et plusieurs exemples significatifs d'exécution de votre programme.

Supposons que le binôme Juliette Capulet et Roméo Montaigu ont réalisé le projet en binôme. Ils créeront un dossier `Capulet_Montaigu` contenant le rapport `Capulet_Montaigu.pdf`, un dossier `src` contenant les textes sources du programme à compiler (voir ci-dessus) et le fichier exécutable, les images traitées (`chaton.png`) et les images réduites (`chaton_resized_20_50.png, chaton_resized_50_20.png, ...`)
N'oublions pas le dossier `Coccinelle` qui contiendra le programme `Coccinelle.java` avec une trace de son exécution et l'exécutable `Coccinelle.class`. Roméo comprimera le dossier `Capulet_Montaigu` en une archive `Capulet_Montaigu.zip` que Juliette enverra avant le 10 juin.