

Analyse d'Algorithmes et Programmation

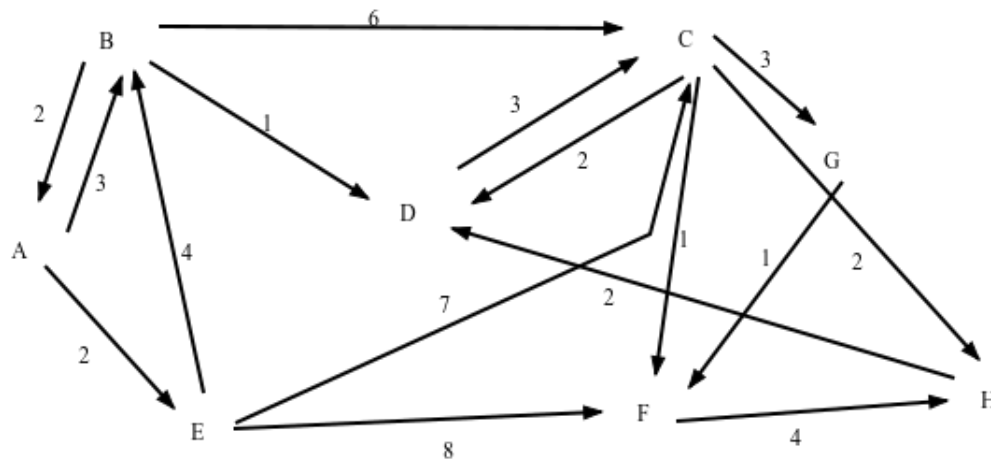
Contrôle Continu 2

Christina Boura et Yann ROTELLA
{christina.boura, yann.rotella}@uvsq.fr

27 avril 2022

1 Graphes

Exercice On considère le graphe suivant.



- (a) Donner les composantes fortement connexes du graphe. Quel algorithme vu en cours, sensiblement modifié peut donner les composantes fortement connexes ? Justifier. (1 point)
- (b) Représenter ce graphe sous forme de matrice d'adjacence. (1 point)
- (c) Combien de bits sont nécessaires à la description de ce graphe (sous forme de matrice d'adjacence) ? Justifier précisément. (1 point)
- (d) Appliquer l'algorithme de Dijkstra sur ce graphe en partant du sommet A. Détailler les étapes de l'algorithme. (1.5 points)
- (e) On suppose maintenant qu'on a un graphe avec une valuation constante (toutes les arêtes sont de poids 1 par exemple). Donner alors une amélioration (en terme de complexité) de l'algorithme de Dijkstra, qui calcule la même chose, mais en un coût plus faible que le Dijkstra originel. (1.5 points)

2 Programmation linéaire

On considère le programme linéaire suivant :

$$\begin{array}{llllll} \text{Maximiser} & & 7x_1 & + & x_2 & \\ & & & & & \\ \text{sous les contraintes} & 4x_1 & + & 3x_2 & \leq & 3 \\ & -x_1 & + & 2x_2 & \geq & -4 \\ & -5x_1 & - & 2x_2 & \leq & 3 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

1. Mettre le programme linéaire sous forme standard.

(0.5 points)

2. Trouver la solution du programme linéaire en détaillant les étapes de l'algorithme du simplexe.

(3 points)

3 Recherche de chaînes de caractères

Construire l'automate de recherche du motif $P = \text{acbabcb}$ en donnant la fonction de transition ainsi que son digramme de transition.

(2 points)

4 Programmation

Un palindrome est un mot qu'on peut lire indifféremment de gauche à droite ou de droite à gauche. Quelques exemples : **pop**, **elle**, **kayak**, **radar**.

Le but de cette partie est de programmer un algorithme de recherche du plus long palindrome dans une chaîne de caractères donnée. Par exemple, le plus long palindrome contenu dans la chaîne **unradar** est **radar**, alors que le plus long palindrome dans la chaîne **cooperative** est **oo**.

Algorithme naïf

1. Programmer une fonction `estPalindrome(s)` qui prend en entrée une chaîne `s` et renvoie `True` si `s` est un palindrome et `False` sinon.

(1 point)

2. Implanter en **Python** un algorithme naïf pour résoudre ce problème. Cet algorithme, va examiner toutes les sous-chaînes possibles, afin de déterminer la plus longue parmi elles qui serait un palindrome.

(3 points)

3. Quelle est la complexité de cet algorithme ? Donner la réponse dans un commentaire en bas de votre code.

(1 point)

Une meilleure approche Nous allons maintenant nous concentrer sur les palindromes de longueur impaire et utiliser, pour résoudre le problème du plus long palindrome, une approche différente basée sur la remarque suivante :

Chaque palindrome de longueur impaire est formé autour d'une lettre centrale qui agit comme miroir. Par exemple **kayak** est un palindrome autour de la lettre **y** et **été** est un palindrome autour de la lettre **t**.

1. Écrire une fonction **eteindre(s, i)** qui prend en entrée une chaîne de caractères **s**, un indice **i** (les indices commencent par 0) et qui renvoie le plus grand palindrome de longueur impaire qu'on peut construire autour du caractère à la position **i** de la chaîne **s**.

Remarque : un caractère peut toujours être considéré comme un palindrome de longueur 1. Cette fonction va donc renvoyer dans tous les cas une chaîne non-vide. Par exemple pour **s = unkayak** et **i = 4**, la fonction **eteindre** doit renvoyer **kayak** alors que pour **i=1** elle doit renvoyer **n**.

(2 points)

2. Écrire une fonction qui prend en entrée une chaîne de caractères **s**, parcourt **s** caractère par caractère, cherche et renvoie le plus long palindrome de longueur impaire présent.

(2 points)

3. Quelle est la complexité globale de cet algorithme (étapes 1+2 comprises). Écrire votre réponse en commentaire de votre code.

(1 point)