

Informatique
Série - Révision

Exercice 1 Zéros consécutifs

Soit un entier naturel n non nul et une liste L de longueur n dont les termes valent 0 ou 1. Le but de cet exercice est de trouver le nombre maximal de 0 contigus dans L (c'est-à-dire figurant dans des cases consécutives). Par exemple, le nombre maximal de zéros contigus de la liste L suivante vaut 4 :

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
L[i]	0	1	1	1	0	0	0	1	0	1	1	0	0	0	0

1. Écrire une fonction $\text{nombreZeros}(L, i)$, prenant en paramètres une liste L , de longueur n , et un indice i compris entre 0 et $n - 1$, et renvoyant :

$$\begin{cases} 0, \text{ si } L[i] = 1 \\ \text{le nombre de zéros consécutifs dans } L \text{ à partir de } L[i] \text{ inclus, si } L[i] = 0 \end{cases}$$

Par exemple, les appels $\text{nombreZeros}(L, 4)$, $\text{nombreZeros}(L, 1)$ et $\text{nombreZeros}(L, 8)$ renvoient respectivement les valeurs 3, 0 et 1.

2. En déduire une fonction $\text{nombreZerosMax}(L)$, de paramètre L , renvoyant le nombre maximal de 0 contigus d'une liste L non vide. On utilisera la fonction nombreZeros .

Exercice 2 Graphe

On considère le graphe G suivant (voir figure 1) composé de 5 sommets reliés par des arcs pondérés par des distances (valeurs entières).

1. Écrire une fonction construireGraphe qui construit et retourne la matrice (M_{ij}) matrice de distances du graphe G , définie par : pour tous les indices i, j , $0 \leq i, j \leq 4$, $M_{i,j}$ représente la distance entre les sommets i et j , ou encore la longueur de l'arête reliant les sommets i et j . On convient que, lorsque les sommets ne sont pas reliés, cette distance vaut -1 . La distance du sommet i à lui-même est, bien sûr, égale à 0.

NB : La matrice des distances du graphe G est un tableau *numpy* à deux dimensions.

2. Écrire une fonction voisins , qui accepte comme paramètres la matrice des distances et un sommet i et retourne la liste des voisins du sommet i .
3. Écrire une fonction degre , qui accepte comme paramètres la matrice des distances et un sommet i et retourne le nombre des voisins du sommet i , c'est-à-dire le nombre d'arêtes issues de i .

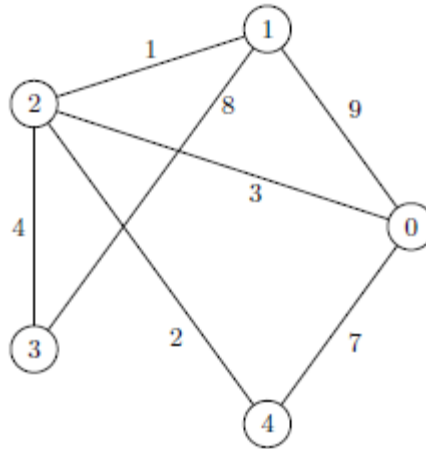


FIGURE 1 – Le graphe G .

4. Écrire une fonction *longueur*, de paramètres la matrice de distance et une liste L de sommets de G , renvoyant la longueur du trajet décrit par cette liste L , c'est-à-dire la somme des longueurs des arêtes empruntées. Si le trajet n'est pas possible, la fonction renverra -1 .

Exercice 3 Jeu du Pendu

Le but de cet exercice est d'implémenter le jeu du pendu. Le principe de ce jeu consiste à découvrir un mot en devinant ses lettres. Dans ce qui suit, nous donnons les étapes de réalisation du script.

1. On initialise une liste intitulée *liste_deviner* par des mots (chaînes de caractères).
2. On choisit aléatoirement un mot *mot_deviner* de cette liste qu'on va cacher à l'utilisateur (le joueur).
3. On initialise une liste intitulée *mot_etat* par n caractères '_'. n étant le nombre de caractères du mot à deviner, qu'on affichera à l'utilisateur au fur et à mesure qu'il devine les lettres.
NB : *mot_etat* est une liste composée de '_' et non pas une chaîne de caractères.
4. L'utilisateur va essayer de découvrir *mot_deviner* en devinant ses lettres. Si la lettre saisie par l'utilisateur figure dans *mot_deviner* alors on affichera la liste *mot_etat* en insérant la lettre devinée dans sa bonne position.
5. L'utilisateur a 10 essais pour deviner le mot. Sinon, il perd le jeu.

Par exemple, si *mot_deviner* est 'rouge', alors on affichera :

Essai 1 : ['_', '_', '_', '_', '_'] - Veuillez saisir une lettre :

Si l'utilisateur saisie la lettre 'z', alors on affichera :

Essai 2 : ['_', '_', '_', '_', '_'] - Veuillez saisir une lettre :

Si l'utilisateur saisie la lettre 'u', alors on affichera : Essai 3 : ['_', '_', '_', 'u', '_'] - Veuillez saisir une lettre : etc

NB : On suppose que tous les mots présents dans *liste_deviner* contiennent des caractères distincts.