

Épreuves écrites NSI
Corrigé du sujet
Amérique du Nord 2022 Jour 1

Exercice 2

1. On lit les tables de routage des différents routeurs pour la bonne destination :

* Du routeur R2 vers le routeur R7 : R2 - R1 - R4 - R7

* Du routeur R7 vers le routeur R2 : R7 - R4 - R3 - R2

2. (a) En cas de panne du routeur R4, le groupe formé par les routeurs R1, R2 et R3 n'aurait plus de connexion avec le du groupe formé par les routeurs R5, R6 et R7.

(b) On peut résoudre ce problème en créant une liaison entre un routeur d'un groupe et un routeur de l'autre groupe : par exemple, entre le routeur R1 et le routeur R6.

3. (a) On peut écrire la table de routage suivante pour R8 :

Table de routage de R8		
Destination	Lien	Distance
R1	R2	2
R2	R2	1
R3	R2	2
R4	R6	2
R5	R6	2
R6	R6	1
R7	R6	2

(b) La nouvelle table de routage de R2 pourrait être :

Table de routage de R2		
Destination	Lien	Distance
R1	R1	1
R3	R3	1
R4	R1	2
R5	R3	3
R6	R8	2
R7	R1	3
R8	R8	1

4. (a)

* Notons d la bande passante du FastEthernet en bit/s. Puisque le coût est de 1, on a :

$$1 = \frac{10^8}{d}$$

Donc $d = 10^8$ bit/s, soit 100 Mbit/s.

* Puisque la bande passante pour un réseau de type Ethernet est de 10Mbit/s soit 10^7 bit/s.

On en déduit le coût c :

$$c = \frac{10^8}{10^7} = 10$$

Le coût est bien de 10 pour une liaison Ethernet.

(b) Le chemin le moins coûteux pour aller de R2 à R5 est le suivant :

R2 - R3 - R4 - R7 - R6 - R5

avec un coût de $65 + 10 + 1 + 1 + 10 = 87$

En effet,

* pour aller de R2 à R5, on doit nécessairement passer par le routeur R4. Or, le chemin R2 - R3 - R4 est moins coûteux que R2 - R1 - R4.

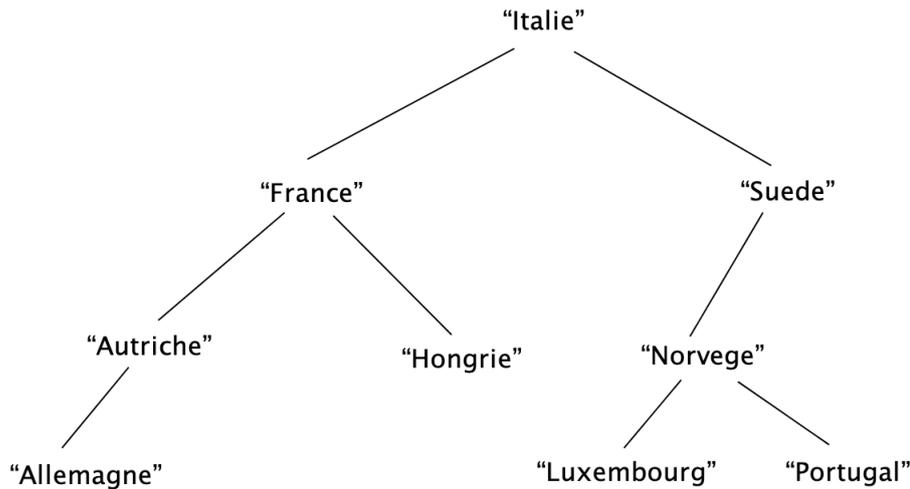
* De plus la liaison directe R4 - R5 (qui est de type E1) est plus coûteuse que le chemin R4 - R7 - R6 - R5 (qui utilise deux liaisons FE et une liaison E).

Exercice 3

1. (a) La hauteur de l'arbre est de 3.

(b) Cette expression est vraie car le caractère 'A' est placé avant le 'P'. Son évaluation en Python donne `True`.

(c) On obtient l'arbre suivant :



2. Le parcours en largeur donne l'ordre de visite suivant :

"Italie" - "France" - "Suede" - "Autriche" - "Hongrie" -
"Norvege"

3. Il faut compléter le code de la façon suivante :

```
def recherche(arb, val):
    """prend pour argument un arbre 'arb' et une valeur
    'val'
    renvoie True si la valeur val est dans l'arbre.
    """
    if est_vide(arb):
        return False
    if val == racine(arb):
        return True
    if val < racine(arb):
        return recherche(gauche(arb), val)
    else :
        return recherche(droite(arb), val)
```

4. On peut écrire le code suivant :

```
def taille(arb):
    if est_vide(arb) :
        return 0
    return 1 + taille(gauche(arb) + taille(droite(arb))
```

Exercice 4

1. (a) Il s'agit de la **proposition 3**. En effet, si la longueur du texte est de 0 ou 1 alors ce texte est un palindrome

(b) On a les valeurs suivantes :

```
* txt[0] : "b"  
* txt[taille - 1] : "r"  
* interieur : "onjou"
```

2. On doit tester un cas où la fonction doit renvoyer True et un autre cas où elle doit renvoyer False. Par exemple

```
* print (palindrome ("laval")) qui doit afficher True ;  
* print (palindrome ("bonjour")) qui doit afficher False.
```

3. On peut écrire la version non récursive suivante :

```
def palindrome(txt) :  
    taille = len(txt)  
    for i in range(taille//2) :  
        if txt[i] != txt[taille-1-i] :  
            return False  
    return True
```

4. a. On peut écrire le code suivant :

```
def complementaire(txt) :  
    res = ""  
    for car in txt :  
        if car == "A" :  
            res = res + "T"  
        elif car == "T" :  
            res = res + "A"  
        elif car == "G" :  
            res = res + "C"  
        elif car == "C" : # ou 'else :'  
            res = res + "G"  
    return res
```

b. Le complémentaire de "GATCGT" est "CTAGCA". Or la chaîne de caractère "GATCGTCTAGCA" (concaténation de ces deux chaînes de caractères) n'est pas un palindrome donc "GATCGT" n'est pas palindromique.

c. On peut écrire le code suivant :

```
def est_palindromique(txt) :  
    nouv_txt = txt + complementaire(txt)  
    return palindrome(nouv_txt)
```

Exercice 5

1. (a) Il s'agit de la proposition 2.

(b) On peut écrire les instructions suivantes :

```
F = creer_file_vide()  
enfiler(F, 15)  
enfiler(F, 17)  
enfiler(F, 14)
```

2. On doit compléter le code de la façon suivante :

```
def longueur_file(F) :
    G = creer_file_vide()
    n = 0
    while not(est_vide(F)) :
        elt = defiler(F)
        enfiler(G, elt)
        n = n+1
    while not(est_vide(G)) :
        elt = defiler(G)
        enfiler(F, elt)
    return
```

3. On doit compléter le code de la façon suivante :

```
def variations(F) :
    taille = longueur_file(F)
    if taille == 1 :
        return []
    else :
        tab = [0 of k in range(taille-1)]
        element1 = defiler(F)
        for i in range(taille-1):
            element2 = defiler(F)
            variation = element2 - element1
            tab.append(variation)
    return tab
```

4. On peut écrire le code suivant :

```
def nombre_baisses(tab) :
    cpt = 0
    mini = 0
    for temp in tab :
        if temp < mini :
            mini = temp
        if temp < 0 :
            cpt = cpt +1
    return (cpt, mini)
```