

## Exercice 1

On rappelle ci-dessous l'algorithme du Tri\_sélection appliqué à un tableau de n nombres .  
On supposera qu'une fonction **Echanger** permet de permuter deux éléments du tableau.

```
1 def tri_selection(tab):
2     for i in range(len(tab)-1):
3         min=i
4         for j in range(i+1,len(tab)):
5             if tab[j]<tab[min]:
6                 min=j
7         echanger(tab,i,min)
8     return tab
9
```

1. Appliquer l'algorithme au tableau de nombres : [30 , 28 , 48 , 12 , 27, 10]

On écrira le contenu du tableau à la fin de chaque itération de la boucle principale **for** (ligne 2).

On complètera le tableau ci-dessous .

|                 |    |    |    |    |    |    |
|-----------------|----|----|----|----|----|----|
| Tableau initial | 30 | 28 | 48 | 12 | 27 | 10 |
| fin itération 1 |    |    |    |    |    |    |
| fin itération 2 |    |    |    |    |    |    |
| fin itération 3 |    |    |    |    |    |    |
| fin itération 4 |    |    |    |    |    |    |
| .....           |    |    |    |    |    |    |
| .....           |    |    |    |    |    |    |
| .....           |    |    |    |    |    |    |

Réponse:

|                 |    |    |    |    |    |    |
|-----------------|----|----|----|----|----|----|
| Tableau initial | 30 | 28 | 48 | 12 | 27 | 10 |
| fin itération 1 | 10 | 28 | 48 | 12 | 27 | 30 |
| fin itération 2 | 10 | 12 | 48 | 28 | 27 | 30 |
| fin itération 3 | 10 | 12 | 27 | 28 | 48 | 30 |
| fin itération 4 | 10 | 12 | 27 | 28 | 48 | 30 |
| fin itération 5 | 10 | 12 | 27 | 28 | 30 | 48 |
| .....           |    |    |    |    |    |    |
| .....           |    |    |    |    |    |    |

2. Combien y a-t-il de comparaisons dans la boucle principale **for** (ligne 2)?

**Réponse :** 15 (5+4+3+2+1)

3. Quelle est la complexité du **tri\_sélection** dans le pire des cas (en fonction de  $n$ , la taille du tableau)? **Réponse :**  $O(n^2)$

4. Dans la fonction **tri\_selection**, les lignes 3 à 6 servent à déterminer l'indice du plus petit élément parmi les éléments dont l'indice est compris entre  $i + 1$  et la fin du tableau.

(a) Ecrire une fonction **mini(tab, i, j)** qui renvoie l'indice du plus petit élément parmi les éléments de **tab** dont l'indice est compris entre  $i$  et  $j$ .

```

1 def mini(tab,i,j):
2     imin = i
3     for k in range(i+1,j+1): #ou range(i,j+1)
4         if tab[k] < tab[imin]:
5             imin = k
6     return imin
7
```

(b) Modifier alors la fonction **tri\_selection** pour utiliser la fonction **mini** à la place des lignes 3 à 6.

```

1 def tri_selection(tab):
2
3     for i in range(len(tab) - 1):
4         .....
5         .....
6     return tab
7
```

```
1 def tri_selection(tab):
2     n = len(tab)
3     for i in range(n-1):
4         imin = mini(tab,i,n-1)
5         echanger(tab,i,imin)
6     return tab
7
8
```

## Exercice 2

**Rappels :** Quelques rappels : Une adresse IPv4 est composée de 4 octets X1.X2.X3.X4 qui peuvent être écrits en notation binaire ou décimale. La notation CIDR X1.X2.X3.X4/n signifie que les n premiers bits de poids forts de l'adresse IP représentent la partie « réseau », les bits suivants la partie " hôte " (machine).

1. (a) Donner le nombre de bits formant un octet.  
  
(b) Déterminer l'écriture décimale de l'adresse IPv4 correspondant à l'écriture binaire :  
11000000.10101000.00000100.11110001
2. On considère la machine d'adresse IPv4 172.20.1.242 / 24.  
  
(a) Donner la notation décimale du masque de sous-réseau de cette machine.  
  
(b) Donner l'adresse décimale de ce réseau.  
  
(c) Donner le nombre maximal de machines que l'on peut connecter sur ce réseau.

## Exercice 2(Corrigé)

**Rappels :** Quelques rappels : Une adresse IPv4 est composée de 4 octets X1.X2.X3.X4 qui peuvent être écrits en notation binaire ou décimale. La notation CIDR X1.X2.X3.X4/n signifie que les n premiers bits de poids forts de l'adresse IP représentent la partie « réseau », les bits suivants la partie " hôte " (machine).

1. (a) Donner le nombre de bits formant un octet.  
*Réponse* : 8 bits  
  
(b) Déterminer l'écriture décimale de l'adresse IPv4 correspondant à l'écriture binaire :  
11000000.10101000.00000100.11110001  
*Réponse* : **192.168.4.241**
2. On considère la machine d'adresse IPv4 172.20.1.242 / 24.  
  
(a) Donner la notation décimale du masque de sous-réseau de cette machine.  
*Réponse*: **255.255.255.0**  
  
(b) Donner l'adresse décimale de ce réseau.  
*Réponse*: **172.20.1.0**  
  
(c) Donner le nombre maximal de machines que l'on peut connecter sur ce réseau.  
*Réponse*: il y a 256 possibilités sur le dernier octet. On enlève l'adresse du réseau et le broadcast. Il reste 254 machines qu'on peut brancher sur le réseau.