

□ **Exercice 1 : Algorithmes de tri**

1. Tri par sélection
  - a) Expliquer en quelques phrases le principe de l'algorithme du tri par sélection
  - b) On applique l'algorithme du tri par sélection à la liste [11, 17, 14, 42, 5, 30, 19]. Donner l'évolution du contenu de la liste au cours des étapes du tri.
2. Tri par insertion
  - a) Expliquer en quelques phrases le principe de l'algorithme du tri par insertion
  - b) On applique l'algorithme du tri par insertion à la liste [11, 17, 14, 42, 5, 30, 19]. Donner l'évolution du contenu de la liste au cours des étapes du tri.
    - ⊗ On pourra donner l'état de la liste après chaque insertion, sans écrire les étapes intermédiaires de ces insertions.

□ **Exercice 2 : Programmation en python**

On veut écrire une fonction `trouve_mini` qui renvoie l'indice de la première occurrence du plus petit élément d'une liste. Par exemple, `trouve_mini([13, 9, 10, 7, 18])` doit renvoyer 3.

1. Que doit renvoyer `trouve_mini([15, 10, 17, 11, 22])` ?
2. Que doit renvoyer `trouve_mini([5, 22, 5, 41, 20, 5, 17])` ?
  - ⊗ On rappelle que lorsque le minimum apparaît plusieurs fois, on doit renvoyer la première occurrence.
3. Compléter le code de la fonction `trouve_mini` ci-dessous.

```
1  ... trouve_mini(liste):  
2  indice_mini = 0  
3  for indice in range(.....):  
4      if ..... < .....:  
5          indice_mini = indice  
6  ..... indice_mini
```

4. Que renverra `trouve_mini([])` ?
5. Modifier cette fonction afin que `trouve_mini([])` renvoie `-1`.

□ **Exercice 3 : Coût**

1. Expliquer rapidement ce que signifie une complexité linéaire pour un algorithme.
2. Donner au moins un exemple, d'algorithme ayant une complexité linéaire.
3. Un algorithme de complexité linéaire traite une liste de 25 000 éléments en 0,03s. Quel est le temps approximatif d'exécution prévisible pour une liste de 150 000 éléments ?
4. On rappelle que : « l'algorithme du tri par sélection à une *complexité quadratique* », expliquer en quelques lignes ce que signifie cette phrase.
5. On a écrit un programme permettant de trier une liste à l'aide du tri par sélection. Si ce programme traite une liste de 5 000 élément en 0,75 secondes, quel est le temps approximatif d'exécution prévisible pour une liste de 80 000 éléments ?