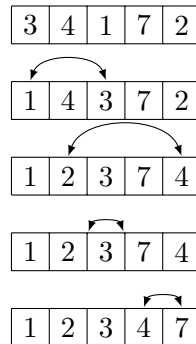


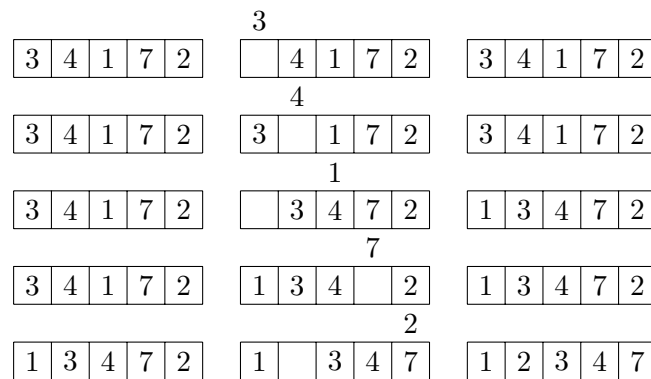
Exercice 1 :

$$\frac{6,8 \times 1000000^2}{16000^2} = 26560s = 7h23min$$

Exercice 2 :



Code 1 – Tri par sélection



Code 2 – Tri par insertion

Exercice 3 : Il faut copier ou importer la fonction `tri_selection`.

```

1 def comparer(tab1: list, tab2: list) -> bool:
2     for i in range(len(tab1)):
3         if not tab1[i] == tab2[i]:
4             return False
5     return True
6
7
8 t1 = [3, 5, 9, 0, 1, 8, 2]
9 t2 = [9, 5, 3, 2, 8, 1, 0]
10 tri_insertion(t1)
11 tri_insertion(t2)
12 print(comparer(t1, t2))

```

Exercice 4 :

```

1 def tri_insertion(tab: list) -> list:
2     """
3     renvoie une copie triée de tab
4     """

```

```
5     tab_trie = []
6     for i in range(len(tab)):
7         # mémoriser
8         en_cours = tab[i]
9         tab_trie.append(en_cours)
10        pos = len(tab_trie)-1
11        # décaler
12        while pos > 0 and en_cours < tab_trie[pos-1]:
13            tab_trie[pos] = tab_trie[pos-1]
14            pos = pos-1
15        # insérer
16        tab_trie[pos] = en_cours
17    return tab_trie
18
19
20 t = [randint(0,100) for _ in range(10)]
```

Exercice 5 :

```
1 def tri_insertion(tab: list) -> None:
2     """
3     tri le tableau de tuples dans l'ordre croissant
4     """
5     for i in range(len(tab)):
6         # mémoriser
7         en_cours = tab[i]
8         pos = i
9         # décaler
10        while pos > 0 and en_cours[0] < tab[pos-1][0]:
11            tab[pos] = tab[pos-1]
12            pos = pos-1
13        # insérer
14        tab[pos] = en_cours
```

Le tri par insertion est un tri stable. Ce n'est pas le cas du tri par sélection.

Exercice 6 :

```
1 def max_occurrences(tab: list) -> int:
2     """
3     renvoie l'élément le plus présent dans le tableau
4     """
5     tri_insertion(tab)
6     en_cours = tab[0]
7     nb_en_cours = 1
8     elt_max = tab[0]
9     nb_max = 1
10    for i in range(1, len(tab)):
11        if en_cours == tab[i]:
12            nb_en_cours += 1
13        else:
14            if nb_en_cours > nb_max:
15                nb_max = nb_en_cours
```

```
16         elt_max = en_cours
17         en_cours = tab[i]
18         nb_en_cours = 1
19     return elt_max
20
21
22 t = [randint(0, 10) for _ in range(100)]
```

Exercice 7 :

```
1 def echanger(tab: list, i: int, j: int) -> None:
2     """
3     échange deux éléments de tab
4     """
5     temp = tab[i]
6     tab[i] = tab[j]
7     tab[j] = temp
8
9
10 def tri_bulles(tab: list) -> None:
11     for i in range(len(tab)):
12         for j in range(1, len(tab)-i):
13             if tab[j-1] > tab[j]:
14                 echanger(tab, j-1, j)
15
16
17 t = [randint(0,1000) for _ in range(20)]
```