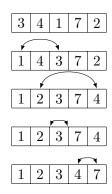
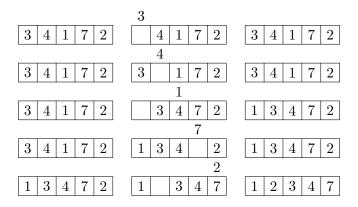
Exercice 1:

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

Exercice 2:



Code 1 – Tri par sélection



Code 2 – Tri par insertion

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

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

Exercice 4:

```
def tri_insertion(tab: list) -> list:
    """
    renvoie une copie triée de tab
    """
```



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

Exercice 5:

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

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

Exercice 6:

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



Exercice 7:

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

