Exercices tris
Correction

Christophe Viroulaud

Première - NSI

Algo 05

Sommaire

1. Exercice 1
2. Exercice 2
3. Exercice 3
4. Exercice 4
5. Exercice 5

Exercice 1

carré du nombre éléments	16000^2	1000000^2
durée	6,8	

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

Sommaire

1. Exercice 1
2. Exercice 2
3. Exercice 3
4. Exercice 4
5. Exercice 5

Exercice 2

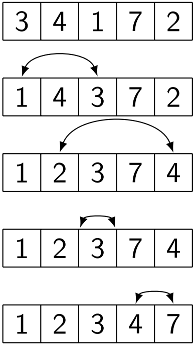
```python
def tri_insertion(tab: list) -> None:
    for i in range(len(tab)):
        # mémoriser
        en_cours = tab[i]
        pos = i
        # décaler
        while pos > 0 and en_cours < tab[pos-1]:
            tab[pos] = tab[pos-1]
            pos = pos-1
        # insérer
        tab[pos] = en_cours
```

```python
tab = [randint(0, 100) for _ in range(10)]
tri_insertion(tab)
print(tab)
```

Sommaire

1. Exercice 1
2. Exercice 2
3. Exercice 3
4. Exercice 4
5. Exercice 5

Exercice 3



3 4 1 7 2

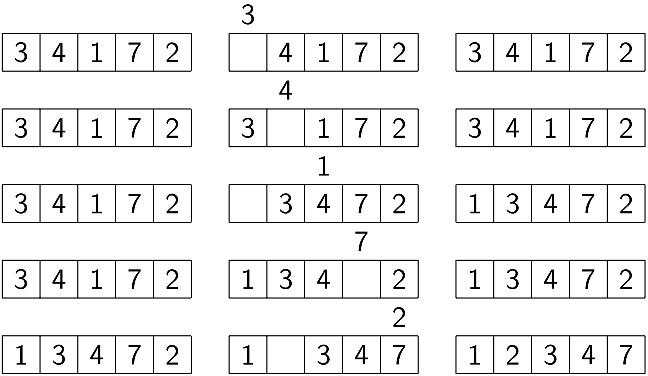
1 4 3 7 2

1 2 3 7 4

1 2 3 7 4

1 2 3 4 7

Code 1 – Tri par sélection



Code 2 – Tri par insertion

Sommaire

1. Exercice 1

2. Exercice 2

3. Exercice 3

4. Exercice 4

5. Exercice 5

Exercice 4

```python
def comparer(tab1: list, tab2: list) -> bool:
    for i in range(len(tab1)):
        if not tab1[i] == tab2[i]:
            # stoppe à la première différence
            return False
    # tous les éléments ont été comparés
    return True
```

```python
t1 = [3, 5, 9, 0, 1, 8, 2]
t2 = [9, 5, 3, 2, 8, 1, 0]
tri_insertion(t1)
tri_insertion(t2)
print(comparer(t1, t2))
```

Sommaire

1. Exercice 1
2. Exercice 2
3. Exercice 3
4. Exercice 4
5. Exercice 5

Exercice 5

```python
def tri_insertion(tab: list) -> list:
    tab_trie = []
    for i in range(len(tab)):
        # mémoriser
        en_cours = tab[i]
        tab_trie.append(en_cours)
        pos = len(tab_trie)-1
        # décaler
        while pos > 0 and en_cours < tab_trie[pos-1]:
            tab_trie[pos] = tab_trie[pos-1]
            pos = pos-1
        # insérer
        tab_trie[pos] = en_cours
    return tab_trie
```

```python
t = [randint(0, 100) for _ in range(10)]
print(tri_insertion(t))
# le tableau initial n'est pas modifié
print(t)
```

Sommaire

1. Exercice 1
2. Exercice 2
3. Exercice 3
4. Exercice 4
5. Exercice 5

Exercice 6