

Exercice 1 Déjà vu...

Rappeler l'implémentation en langage Python des fonctions citées ci-dessous déjà vues en cours ou en TP et déterminer leur complexité temporelle.

1. maxi
2. moyenne
3. variance

Exercice 2 Recherche d'un mot dans un texte

La fonction `RechercheMot(texte,mot)` est documentée de la manière suivante :

```
"""Cette fonction recherche dans un texte la première occurrence d'une suite de caractères
Entrée: texte (type 'str') et mot (type 'str')
Sortie: position (type 'int' ou 'bool')
Si la suite de caractères cherchée (mot) est trouvée, la fonction renvoie la position de la
première lettre du mot cherché
Sinon, position prend la valeur booléenne 'False' """
```

1. Écrire un algorithme réalisant la fonction `RechercheMot`
2. Traduire cet algorithme en langage Python.
3. Quelle est la complexité de cet algorithme ?

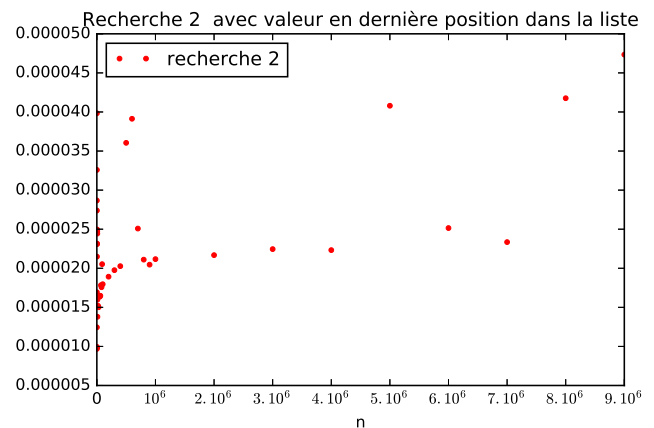
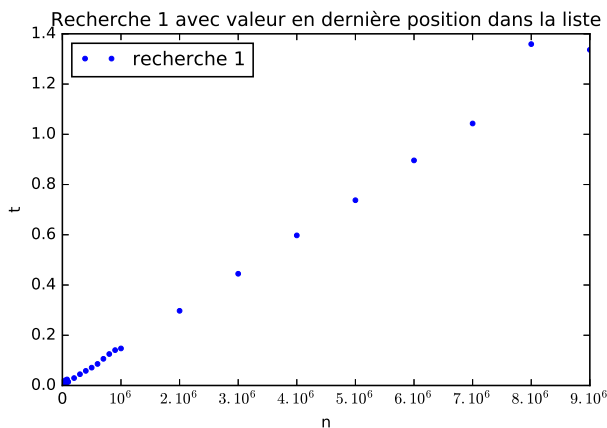
Exercice 3 Comparaison d'algorithmes

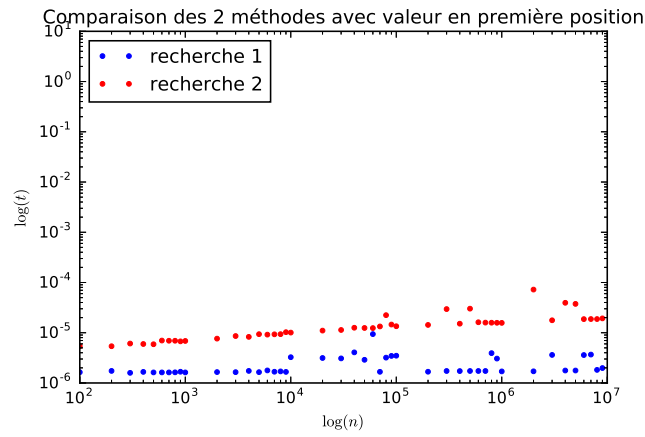
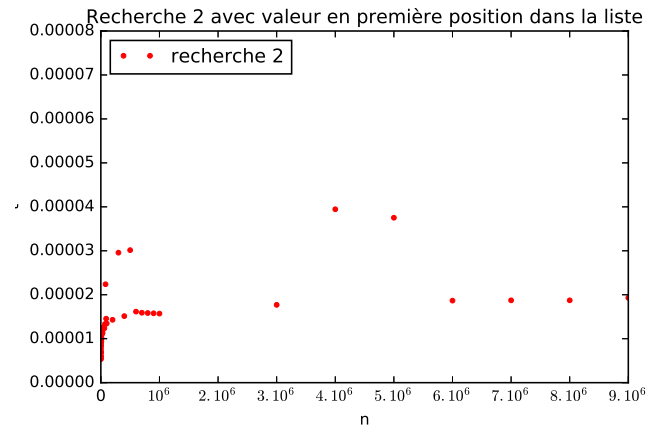
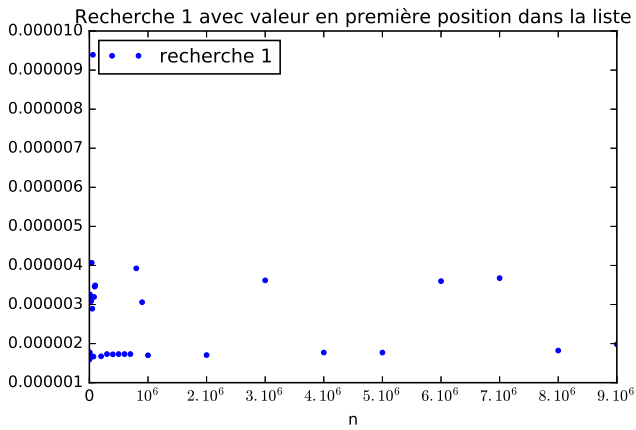
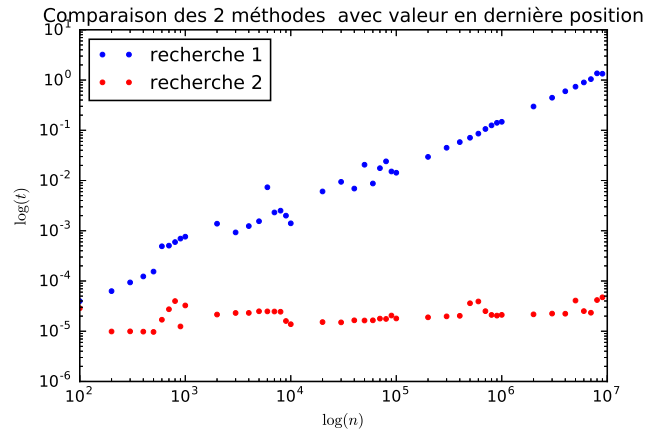
On dispose d'une liste de valeurs numériques triées (dans l'ordre croissant ou décroissant). Deux méthodes de recherches d'une valeur dans une liste triée sont envisagées et désignées par `recherche 1` et `recherche 2`.

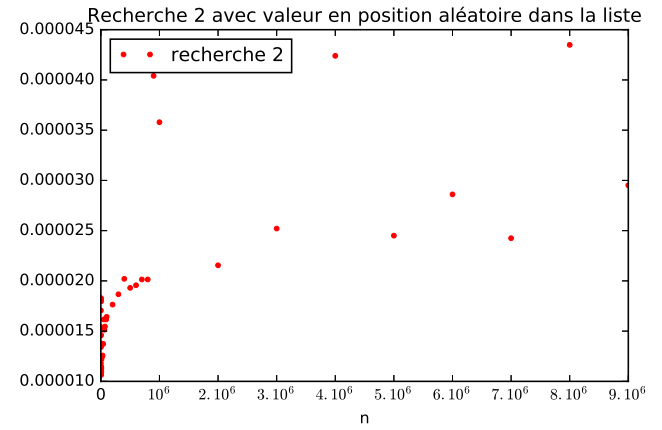
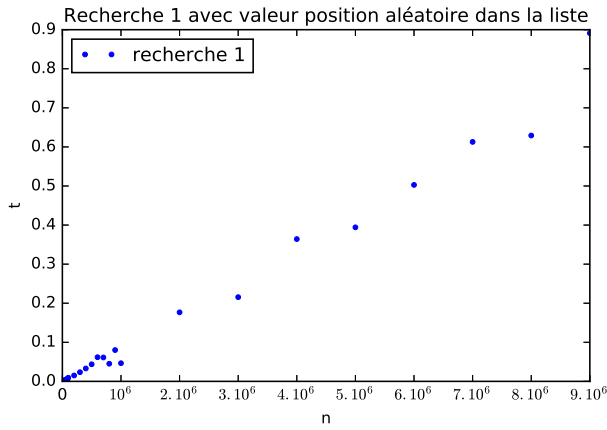
Les durées d'exécution de différentes recherches sont mesurées pour les deux méthodes dans différentes conditions :

- la valeur recherchée est en dernière position dans la liste de taille n ,
- la valeur recherchée est en première position dans la liste de taille n ,
- la valeur recherchée est située aléatoirement dans la liste de taille n .

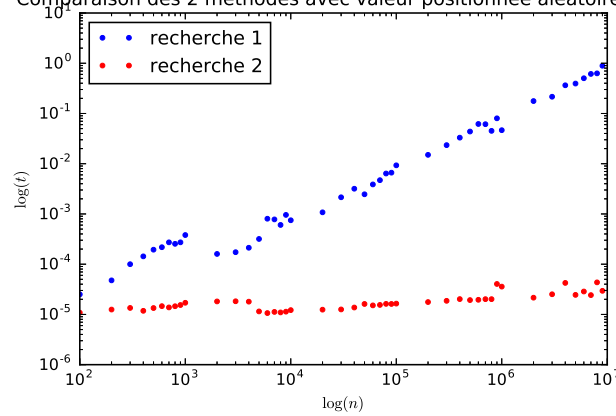
Les résultats expérimentaux sont réunis dans les figures suivantes.







Comparaison des 2 méthodes avec valeur positionnée aléatoire



1. Rappeler comment on peut définir, en langage Python, une fonction de recherche séquentielle d'une valeur dans une liste, qui termine dès que la valeur est trouvée en renvoyant la position de la première occurrence de la valeur cherchée. Si la valeur n'est pas dans la liste la fonction renvoie `False`.
2. Analyser les différentes courbes et identifier laquelle des 2 méthodes correspond à la recherche séquentielle.
3. Comparer et commenter l'évolution de la durée de calcul lors de l'exécution des deux algorithmes dans les différents cas.