

BUT 3 – R5.A.04.Qualité algorithmique  
TP 1 : complexité algorithmique

**Exercice 1.**

Écrire un programme Python qui fait une recherche naïve d'un nombre dans un tableau (liste) initialisé(e) par des nombres compris entre 0 et 100 000 ( $i \% 100000$ ). Le nombre recherché ne doit pas exister (-1, par exemple).

Écrire un autre programme Python qui fait une recherche dichotomique (binaire).

Calculer et afficher le temps d'exécution du premier puis du second pour 100k nombres, puis pour 500k, 750k, 1M, 5M, 7,5M, 10M, 50M, 75M et 100M. Mesurer des moyennes de 10 itérations.

Comparer les courbes des deux implémentations.

Comparer les résultats avec les complexités théoriques de ces deux algorithmes de recherche.

Tester l'initialisation de la liste avec des nombres aléatoires. Que remarquez vous dans le temps d'exécution ?

**Exercice 2.**

Choisir un problème (parcours de tableaux multi-dimensionnels, tri, calcul matriciel, ...) et au moins 2 solutions ayant des classes de complexité différentes. Proposer les implémentations de ces solutions, avec le langage de votre choix.

Fournir une comparaison similaire à l'exercice 1.

Aidez-vous d'outils d'IA générative (ChatGPT par exemple) pour obtenir des implémentations équivalentes, mais dans un langage de programmation différent. Comparez les 2 langages.

Déposer le travail fait sous la forme d'une archive ZIP sur Moodle.