

# Rapport filtre de bloom

## Introduction:

Dans le cadre de la SAE3.08 nous devons implémenter un filtre de Bloom en utilisant plusieurs structures de données et de comparer ces différentes implémentations par un banc d'essai.

Le filtre de bloom permet de savoir l'absence d'un élément (il ne peut pas y avoir de faux négatif ) et avec une certaine probabilité la présence d'un élément ( il peut y avoir des faux positifs )

## Comment j'ai fais:

Tout d'abord j'ai commencé par créer les structures de données (Tableau, ArrayList et LinkedList).

Ensuite j'ai écrit les fonctions de hachage pour chaque structure.

Et pour finir je me suis occupé de toute la partie test y compris le banc d'essai.

## Description des classes:

*FiltreBloomArrayList*, *FiltreBloomLinkedList*, *FiltreBloomTableau* : dans ces trois classes on crée le filtre de bloom pour chaque structure.

Le constructeur crée un filtre comprenant l'ensemble de bits et le nombre de fonctions de hachage.

Et on retrouve des fonctions pour ajouter un élément, fonction de hachage, calcul nombre de bit et calcul de faux positif.

*FiltreDeBloom*: Dans cette classe on retrouve les tests du filtre de bloom pour chaque structure.

## Résultat:

( voir capture )

On peut voir que niveau temps d'exécution le tableau est le plus rapide et la linkedList est largement plus lente.

Au niveau de taux d'erreur, la probabilité d'erreur est plus élevée pour le tableau et linkedList que pour l'arrayList.