

# Expérience

## Objectif

Le but est de mesurer le nombre d'opérations réalisées dans trois algorithmes utilisant des stratégies différentes pour traiter le problème ci-dessous.

Un expérimentateur réalise des expériences sur des marqueurs (par exemple différents échantillons de sang). Chaque expérience permet de tester une hypothèse (par exemple la présence anormale de globules blancs) et fournit un résultat soit positif soit négatif.

Il dispose d'un certain nombre  $m$  de marqueurs et son protocole expérimental lui permet de récupérer la liste des marqueurs ayant fourni une réaction positive à l'expérience menée (et pas ceux ayant un résultat négatif). Le nombre de ces marqueurs "positifs" sera noté  $p$  avec  $p \leq m$ . Le but est de retrouver à partir de ces deux ensembles l'ensemble des marqueurs négatifs.

## Modélisation informatique et code fourni

Les marqueurs sont représentés par des entiers. On peut supposer que chaque marqueur a reçu un identifiant unique. L'ensemble des marqueurs est stockée dans une liste triée par ordre croissant. Les marqueurs positifs sont aussi stockés dans une liste triée par ordre croissant. Les marqueurs négatifs sont donc ceux qui sont dans la première liste mais pas dans la seconde.

Le fichier *expérimentateur.py* disponible sur ARCHE contient une fonction qui permet de générer la liste des marqueurs ainsi que la liste des marqueurs positifs. Les deux listes sont retournées triées par ordre croissant. La fonction prend comme entrées le nombre total de marqueurs  $m$  et le nombre de marqueurs positifs  $p$ .

## Travail à faire

Pour chaque stratégie ci-dessous, il faut écrire une fonction qui prend en paramètre la liste des marqueurs et la liste des marqueurs positifs et qui retourne deux choses :

1. La liste des marqueurs négatifs
2. Le nombre de comparaisons entre deux marqueurs effectués (cela représentera le coût de l'algorithme)

Les deux premières stratégies sont basées sur l'idée de rechercher si un marqueur est dans la liste des marqueurs positifs. S'il n'y est pas, c'est que c'est un marqueur négatif. La troisième stratégie utilise une approche différente. Vous complèterez le fichier `experimentateur.c` qui se trouve sur ARCHE. Vous pouvez cependant rendre votre travail dans le langage de votre choix mais la réimplémentation des fonctions présentent dans le fichier devra être faite.

### Stratégie 1 : utilisation d'une recherche séquentielle

Dans cette stratégie, on utilise une **recherche séquentielle** pour rechercher la présence d'un marqueur dans la liste des marqueurs positifs.

### Stratégie 2 : Utilisation d'une recherche dichotomique

Dans cette stratégie, on utilise une **recherche dichotomique** pour rechercher la présence d'un marqueur dans la liste des marqueurs positifs.

### Stratégie 3 : Stratégie ne reposant pas sur une recherche

Le principe est le suivant. On parcourt dans la liste des marqueurs jusqu'à rencontrer le plus petit marqueur positif. Tous les marqueurs qui sont avant sont des marqueurs négatifs. Ensuite, on continue à parcourir la liste jusqu'à trouver le second plus petit marqueur positif. Les marqueurs passés sont donc aussi des marqueurs négatifs. On répète l'opération jusqu'à ce que l'on ait parcouru tous les marqueurs. Votre fonction doit donc comporter une unique boucle qui parcourt la liste des marqueurs il n'y a pas besoin de faire de recherche séquentielle ou de recherche dichotomique.

## Recherche empirique des cas favorables

Nous allons maintenant réaliser des tests en faisant varier le nombre de marqueurs positifs des expériences pour observer la variation du nombre d'opérations. On utilisera la fonction `experience` qui permet d'obtenir un tableau de marqueurs positifs.

le programme de test pour qu'il permette de produire pour un nombre de marqueurs donné en argument et un nombre de marqueurs positifs d'une expérience variant de 1 au nombre de marqueur, le décompte des opérations pour les 3 versions. Pour  $m = 10$ , votre programme produira une sortie du type (les chiffres donnés ici ne sont pas nécessairement ceux que vous trouverez)

```
10 1 10 10 39
10 2 19 10 41
10 3 27 11 43
10 4 34 12 45
10 5 40 13 53
10 6 45 20 56
10 7 49 24 60
10 8 52 29 65
10 9 54 26 62
10 10 55 38 76
```

où les colonnes correspondent respectivement à  $m$ ,  $p$  et aux décomptes.

Ensuite, vous créerez 10 fichiers pour  $m = 10, 20, 30, 40, 50, 70, 80, 90, 100$  que vous nommerez les fichiers `res- $m$ .dat`. Finalement, vous tracerez des courbes (avec le logiciel de votre choix) à partir des données pour les différentes valeurs de  $m$ .

## Travail à rendre

- Le travail est à faire en binôme.
- Le travail est à rendre sur ARCHE (dimanche 13 mars 23h59)
  - Le code
  - un rapport contenant une analyse de la complexité dans le pire des cas en fonction de  $p$  et  $m$  des trois stratégies
  - les 10 courbes
  - une recommandation sur la meilleure stratégie à utiliser par l'expérimentateur.