

# Exercice sur machine : Expérience

## Complexité

### 1 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  $p$  ( $p \leq m$ ).

### 2 Code fourni

Vous complétez le fichier `experimentateur.c` qui se trouve sur ARCHE. Le fichier contient des fonctions permettant de créer une expérience. Il s'agit d'une structure qui contient le tableau des marqueurs ayant servi à l'expérience ainsi que le tableau contenant les marqueurs positifs.

Les marqueurs sont représentés par des entiers. On peut supposer que chaque marqueur a reçu un identifiant unique.

Une fonction pour trier un tableau d'entiers est aussi fournie.

Une fonction test qui prend en paramètre le nombre de marqueurs ayant fourni une réponse positive  $p$  et le nombre total de marqueurs. Ce programme est fonctionnel mais il ne calcule pas ce qui est demandé dans cet exercice.

### 3 Stratégie 1

Dans une première phase, l'expérimentateur, qui ne connaît pas grand chose à l'informatique, utilise une approche naïve qui consiste pour chacun des  $m$  marqueurs à vérifier si le marqueur se trouve dans le tableau des marqueurs positifs de manière itérative (il s'arrête tout de même dès qu'il a trouvé). Il ne dispose pas de la possibilité de modifier ce tableau.

1. Proposez une première solution implantant cette stratégie. Ecrivez le code de la fonction `marqueurs_negatifs1` retournant un tableau contenant les marqueurs négatifs. Testez son bon fonctionnement.
2. Pour tester la complexité de cet algorithme, quelle opération proposez-vous de compter ? (Cette opération sera désignée par OP)
3. Existe-t'il un pire des cas pour cet algorithme ? Si oui le décrire, sinon justifier.

4. Exprimer  $c_1(m, p)$  le nombre exact d'opérations OP (pour le pire des cas ou le cas général suivant votre réponse à la question précédente) en fonction de  $m$  et  $p$ .

## 4 Stratégie 2

L'expérimentateur rencontre un étudiant qui s'y connaît un peu plus en informatique et qui fait la suggestion suivante : *"Il est possible de trier les marqueurs positifs avant de chercher dedans. Une fois triée, il est même possible de faire quelque chose pour chercher plus vite, mais je ne me souviens plus quoi"*.

1. Proposer une seconde solution où le tableau des marqueurs positifs est trié. Ecrivez le code dans la fonction `marqueurs_negatifs2` et testez là. Essayez de trouver la solution la plus efficace en termes de complexité.
2. Existe-t'il un pire des cas pour cet algorithme ? Si oui le décrire, sinon justifier.
3. Trouvez une borne supérieure pour le nombre d'opérations  $c_2(m, p)$  en fonction de  $m$  et de  $p$ .

## 5 Stratégie 3

Un autre étudiant ajoute : *"Mon camarade a raison, mais on pourrait aller encore plus vite en triant à la fois les marqueurs positifs et les marqueurs."*

1. Proposez une troisième solution où à la fois les marqueurs ET les marqueurs positifs sont triés. Ecrivez le code de la fonction `marqueurs_negatifs3`. Testez.
2. Existe-t'il un pire des cas pour cet algorithme ? Si oui le décrire, sinon justifier. Trouver une borne supérieure pour le nombre d'opérations  $c_3(m, p)$  en fonction de  $m$  et de  $p$ .

## 6 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.

1. Modifiez vos codes pour que l'on puisse compter le nombre d'opérations OP effectuées dans chacune des trois versions.
2. Essayez de déterminer, pour chaque version, un exemple du problème (les valeurs de  $m$  et  $p$  suffiront) pour lequel une des versions est plus performante. Que constatez-vous ? Expliquez.
3. Modifiez 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.

4. Créez 10 fichiers pour  $m = 10, 20, 30, 40, 50, 70, 80, 90, 100$ . Vous nommerez les fichiers `res- $m$ .dat`.
5. Tracer des courbes (avec le logiciel de votre choix) à partir des données pour les différentes valeurs de  $m$ .

## 7 Travail à rendre

- Le travail est à faire en binôme.
- Le travail est à rendre sur ARCHE pour le jeudi 11 mars 2021 à 18h. Vous déposerez :
  1. le fichier `experimentateur.c`
  2. les 10 fichiers `res- $m$ .dat`
  3. un rapport contenant vos réponses aux questions du sujet ainsi que les 10 courbes