SAE 1.02 - E3CETE

J. Renaud - M. Franceus-Cointrel

Pour le 14 janvier 2024

Table des matières

Introduction

Dans cette SAE nous étudions...

Chapitre 1

Analyse et comparaison des 3 méthodes de tris

1.1 Les fonctions de tris et comptage du nombre d'opérations approximatif

1.1.1 Tri par sélection

```
%insert code here
```

Code 1.1: *Focntion marche aléatoire*

1.1.2 Tri par bulles

1.1.3 Tri par insertion

1.2 Protocole de test

1.2.1 Définition des variables

L'objectif de cette expérience est d'évaluer la performance de chacun des tris. Pour cela nous devons réaliser chacun des tris en variant le nombre de cartes N contenus dans le Paquet. Pour cette expérience les paquets triés seront toujours pleins, et le nombre de cartes N contenus est déterminé par les cardinalités des caractéristiques possibles. En principe nous fixons les cardinalités associés aux couleurs/figures/textures pour ne seulement varier le nombres de répétition des figures. Ainsi N est défini tel que :

$$N = cardCouleurs \times cardFigures \times cardTextures \times cardRepetFigures$$

Pour chacun des méthodes de tris nous calculons le nombre d'opération $nOpApprox$ et son temps d'éxecution (en ms) $tempsExec$. Ainsi, en répétant les tris $nbRepetTest$ nombre de fois pour N fixé, nous en déduisions les valeurs moyennes ainsi que leurs incertitudes (écartype) telles que :

$$\boldsymbol{nbOpMoy = \frac{1}{nbRepetTest} \times \sum_{i=1}^{nbRepetTest} nbOpApprox_i} \tag{1.1}$$

$$\boldsymbol{u\_nbOp = \sqrt{\frac{1}{nbRepetTest} \times \sum_{i=1}^{nbRepetTest} (nbOpApprox_i - nbOpMoy)^2}} \tag{1.2}$$

$$\boldsymbol{tempsExecMoy = \frac{1}{nbRepetTest} \times \sum_{i=1}^{nbRepetTest} tempsExec_i} \tag{1.3}$$

$$\boldsymbol{u\_tempsExec = \sqrt{\frac{1}{nbRepetTest} \times \sum_{i=1}^{nbRepetTest} (tempsExec_i - tempsExecMoy)^2}} \tag{1.4}$$

1.2.2 Protocole expérimental

1. Réaliser les 3 tris sur $nbRepetTest$ paquets ayant les mêmes caractéristiques mais chacun mélangés différemment, pour un même nombre de cartes N.
2. Récupérer le nombre d'opération $nbOpApprox$ et le temps d'éxécution $tempsExec$ de chacun des tris pour toutes les répétitions.
3. Calculer et stocker les valeurs moyennes $nbOpMoy$ (1.1) et $tempsExecMoy$ (1.3), ainsi que leurs incertitudes u_nbOp (1.2) et $u\_tempsExec$ (1.4).
4. Répéter l'expérience en variant N, en modifiant la valeur de $cardRepetFigures$.

1.2.3 Fonctions Test

```
%insert code here
```

Code 1.2: *Focntion marche aléatoire*

1.3 Test n°1

1.3.1 Initialisation et spécificités

Les variables :

- int $nbRepetTest = 1000$
- int $cardCouleurs = 1$
- int $cardFigures = 1$
- int $cardTextures = 1$
- int $cardRepetFigures \in [10 - 500]$

Specificités de l'appareil utilisés : ....

1.3.2 Analyse graphique

Nous stockons les données des tests sous forme de 3 fichiers .csv, grâce à une fonction de conversion. Chaque fichier correspond à une méthode de tri et contient toutes les valeurs de N, *nbOpMoy*, *u_nbOp*, *tempsExec*, *u_tempsExec* associées à cette dernière. Nous choisissons de réaliser le graphique de $nbOpMoy = f(N)$ et de $tempsExecMoy = f(N)$ à l'aide du logiciel *Regressi*.

Etant donées que ces 3 fonctions de tris utilisent chacune une double boucle imbriquée, nous nous attendons à une complexité quadratique $O(n^2)$.

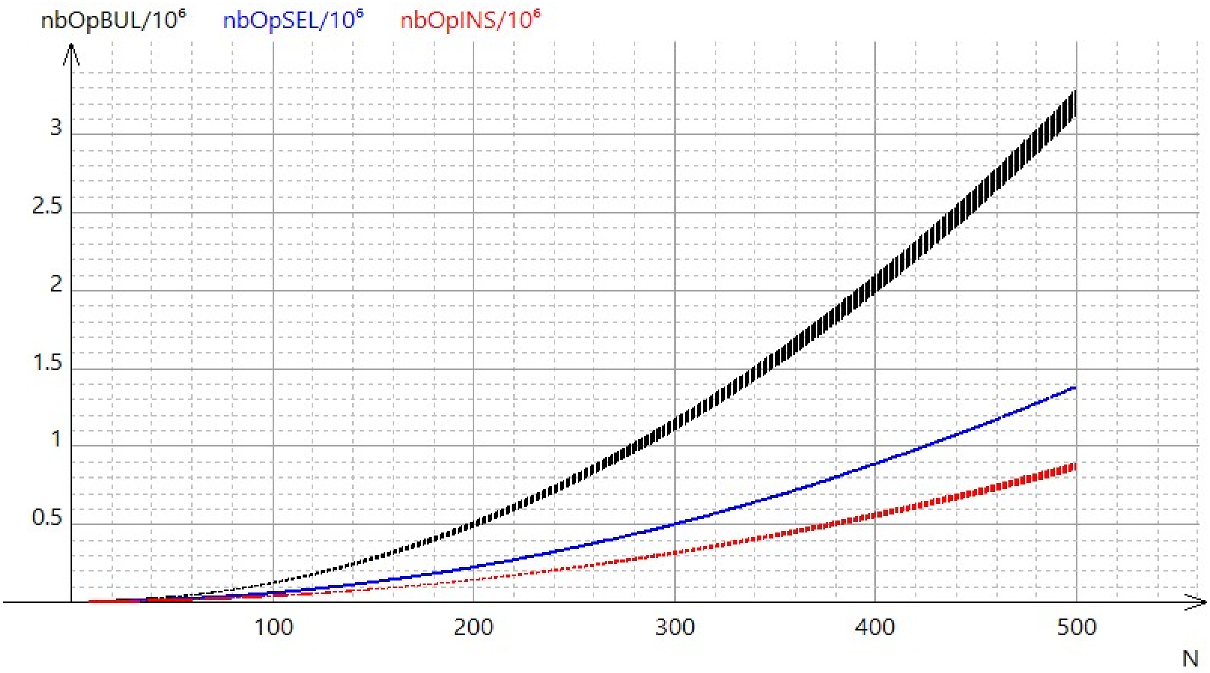

Figure 1.1: Graphique de $nbOpMoy = f(N)$ avec les barres d'incertitudes sur *nbOpMoy* pour les 3 méthodes de tris : en bleu ....

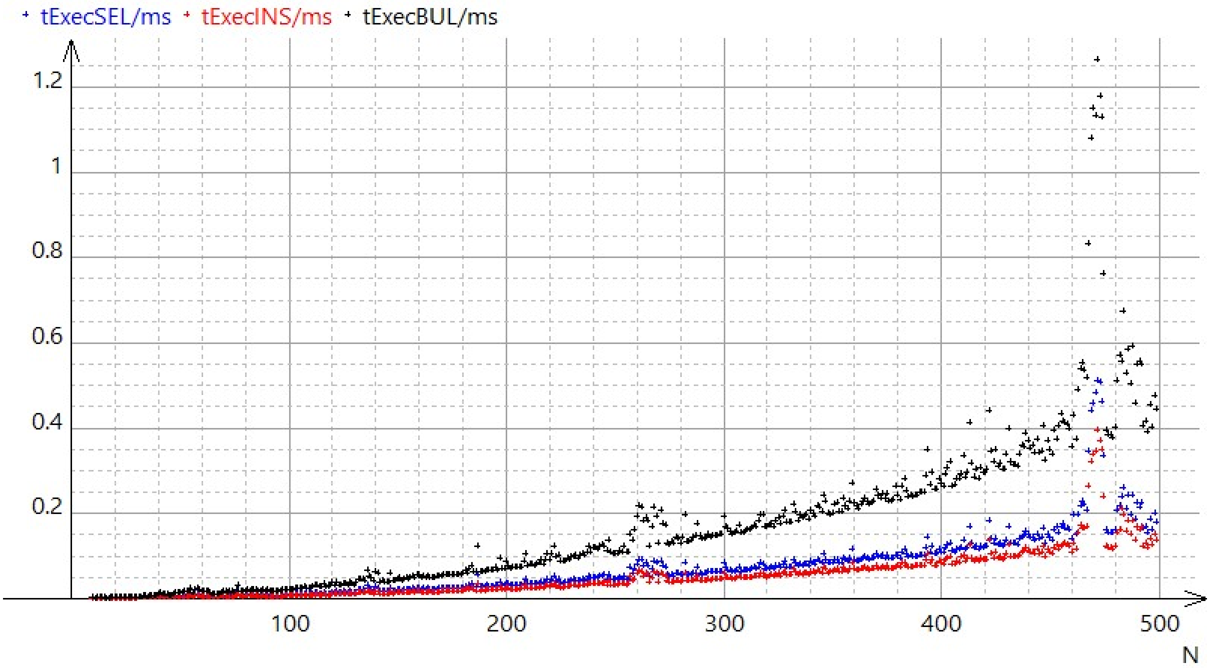

Figure 1.2: Graphique de $tempsExecMoy = f(N)$ pour les 3 méthodes de tris : en bleu ....

Commentaire : ....

Chapitre 2

Théorie

2.1 Représentation mathématique de la Class *Table*

1. Une *Table* est une liste ordonée de *Cartes* tous différentes (pas de répétition), parmi toutes les *Cartes* du jeu contenues dans un *Paquet*. Une *Table* est donc un arrangement.

2. Pour une *Table* de 9 *Cartes* et un jeu de 81 *Cartes*, le nombre de *Tables* différentes possibles est tel que :

$$A_{81}^{9} = \frac{81!}{(81-9)!} = 81 \times 80 \times ... \times 74 \times 73 = 94670977328928000$$

2.2 Etude du cas 3CR

2.2.1 Calculs théoriques

3. Sachant que pour une *Carte* il y a 3 *Couleurs*, 3 répétitions maximales de figures, 3 *Figures* et 3 *Textures* possibles ; on en déduit qu'il existe $1 \times 3 \times 3 \times 3 = 27$ cartes rouges distinctes. Pour déterminer les arrangements possibles contenant exactement 3 cartes rouges, nous comptons le nombre de combinaisons possibles de 3 cartes rouges parmi les 27 que multiplie le nombre de combinaisons possibles des 6 cartes restantes de la *Table* parmi les 54 cartes non rouges. A chacune de ces combinaisons possibles de 3 cartes rouges + 6 cartes non rouges (pas d'ordre), nous pouvons réaliser 9! permutations possibles. Ainsi, le nombre de *Tables* différentes contenant exactement 3 *Cartes* rouges est tel que :

$$C_{27}^{3} \times C_{81-27}^{6} \times 9! = \frac{27!}{3!(27-3)!} \times \frac{54!}{6!(54-6)!} \times 9! = 2925 \times 25827165 \times 362880 = 27413572782960000$$

4. On en déduit la probabilité P_{3CR} d'obtenir une *Table* contenant exactement 3 cartes rouges :

$$P_{3CR} = \frac{casFavorables}{casTotal} = \frac{C_{27}^{3} \times C_{81-27}^{6} \times 9!}{A_{81}^{9}} = \frac{27413572782960000}{94670977328928000} = 0.28956680871386137$$

2.2.2 Fonction *proba3RC*

5. fonction here

2.2.3 Traitements de données et analyse graphique

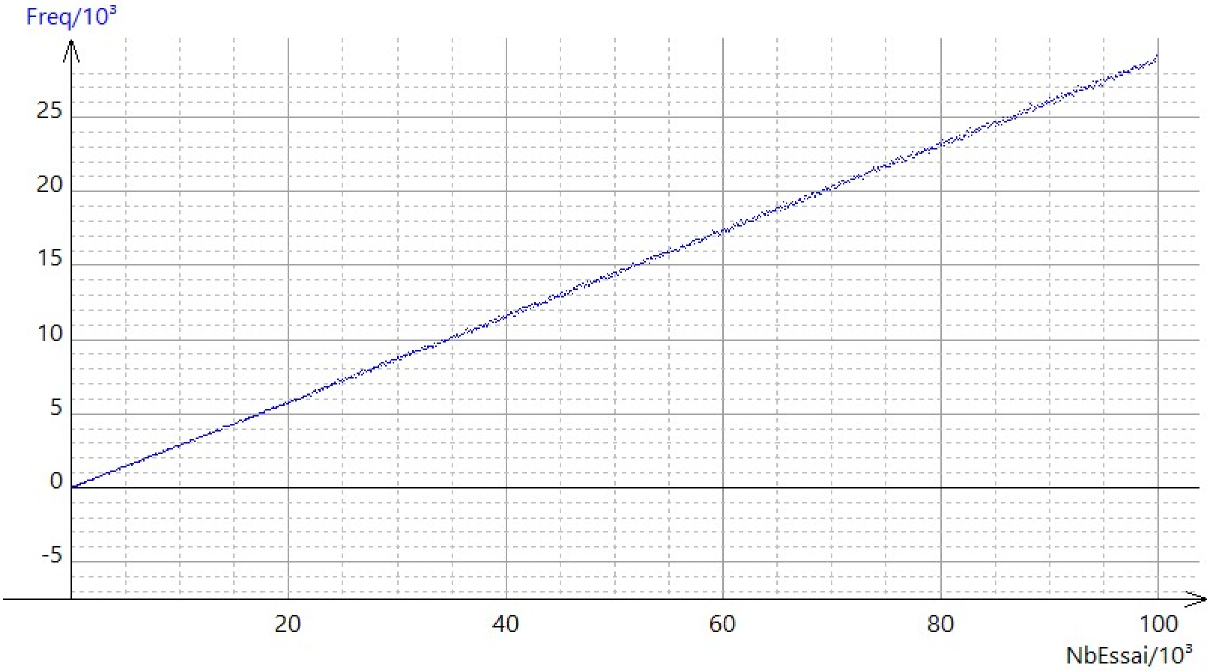

Figure 2.1: Graphique de $Freq = f(nbEssai)$ en faisant varier $nbEssai$ de 100 à 100 000 par pas de 100

Les graphiques obtenus à partir des données générées par cette fonction permettent de déterminer une valeur expérimental de $P_{3CR}exp$. En effet, en traçant $Freq = f(nbEssai)$ nous apercevons que la courbe varie linéairement. Ceci est en accord avec la théorie puisque $P_{3CR}theo = \frac{casFavorables}{casTotal}$ soit $casFavorables = P_{3CR} \times casTotal$ ce qui équivaut à $Freq = a \times nbEssai$. En réalisant une modélisation linéaire grâce à l'outils de modélisation sur le logiciel Regressi ; nous en déduisons :

$$P_{3CR}exp = a = 0.28939 \pm 0.00015$$

Nous pouvons calculer le pourcentage d'erreur Err telle que :

$$Err = abs\left(\frac{P_{3CR}theo - P_{3CR}exp}{P_{3CR}theo}\right) \times 100 = 0.061\%$$

6. fonction here

   Pour déterminer à partir de combien d'essais il faut pour obtenir un résultat fiable, nous avons choisi de tracer $Err = f(nbEssai)$ à partir des probabilités propres à chacun des événements (associé à nombre d'essais) : $P = Freq/nbEssai$.

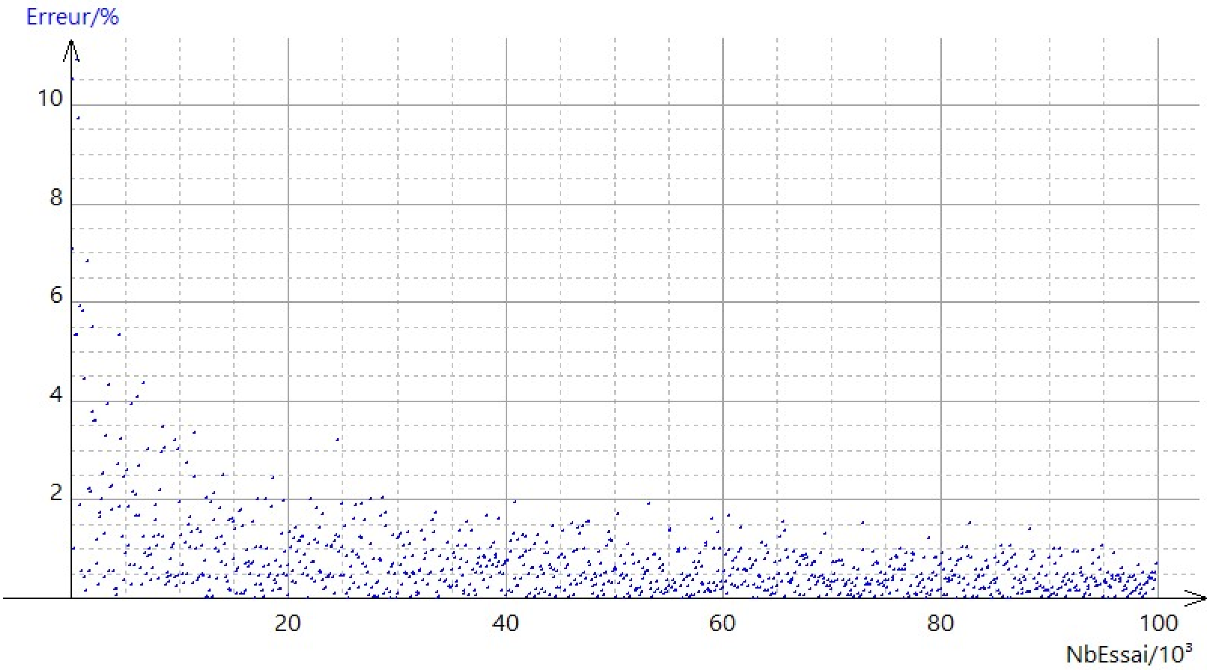

Figure 2.2: Graphique de $Err = f(nbEssai)$ en faisant varier $nbEssai$ de 100 à 100 000 par pas de 100

Nous observons que pour avoir un résultat fiable à 5% d'erreur près, il faut au moins environ 10 000 essais. Le graphique ci-dessous de $P = f(nbEssai)$ illustre bien la répartition des probabibilités qui converge vers la valeur théorique en augmentant le nombre d'essais.

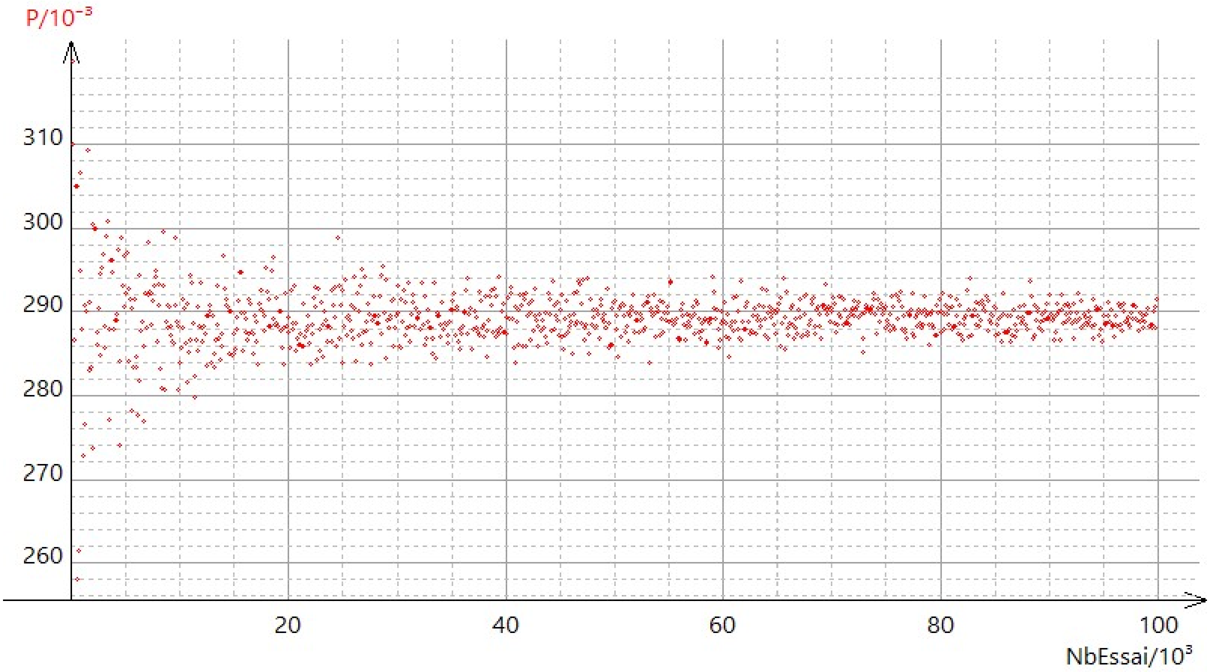

Figure 2.3: Graphique de $P = f(nbEssai)$ en faisant varier $nbEssai$ de 100 à 100 000 par pas de 100

2.3 Etude du cas 3CR&2CL

2.3.1 Caluls théoriques

7. L'intersection de l'ensemble des cartes rouges et des cartes ayant au moins un losange n'est pas vide. Il faut étudier chacune des cas. Il y a 27 cartes rouges distincts et 27 cartes ayant au moins un losange. Parmi ces cartes rouges seulement $1 \times 3 \times 2 \times 3 = 18$ n'ont pas de losange. De même, parmi les cartes ayant au moins un losange, $2 \times 3 \times 1 \times 3 = 18$ ne sont pas rouges. Ainsi, $1 \times 3 \times 1 \times 3 = 9$ cartes sont à la fois rouges et ont au moins un losange.

Table 2.1: Tableau récapitulatif des différents cas possibles pour obtenir une combinaison de 9 cartes avec exactement 3 cartes rouges et 2 cartes ayant au moins un losange :

On en déduit le nombre d'arrangements possibles de *Table* contenant exactement 3 cartes rouges et 2 cartes ayant au moins 1 losange tel que :

$$
\begin{aligned}
&9! \times (\, C_9^2 \times C_{18}^1 \times C_{18}^0 \times C_{36}^6 \\
&\quad + C_9^1 \times C_{18}^2 \times C_{18}^1 \times C_{36}^5 \\
&\quad + C_9^0 \times C_{18}^3 \times C_{18}^2 \times C_{36}^4) \\
&\quad = 362880 \times 17960464368 = 6517493309859840
\end{aligned}
$$

On en déduit la probabilité $P_{3CR\&2CL}$ de cet événement :

$$P_{3CR\&2CL} = \frac{casFavorables}{casTotal} = \frac{6517493309859840}{94670977328928000} = 0.06884362550959248$$

2.3.2 Vérification empirique succincte

8. fonction here

2.4 Etude du cas E3C

2.4.1 Methode *estUnE3C*

2.4.2 Traitements de données et analyse graphique

Annexe A

Modules utilisés

```
import math
  from math import pi
import random as rnd
import statistics as stats
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.cm as cm
  from matplotlib.patches import Ellipse
  from mpl_toolkits import mplot3d
import numpy as np
import scipy as scp
import scipy.stats as st
  from scipy.stats import multivariate_normal
```

Code A.1: *Modules utilisés en Python 3.9.2*

Annexe B

Sources

- https://moodle.umontpellier.fr/course/view.php?id=25363
- https://femto-physique.fr/physique_statistique/diffusion-moleculaire.php
- https://stringfixer.com/fr/Random_walk
- https://en.wikipedia.org/wiki/Mass_diffusivity
- https://fr.wikipedia.org/wiki/Mouvement_brownien
- https://fr.wikipedia.org/wiki/Lois_de_Fick
- https://fr.wikipedia.org/wiki/Cha%C3%AEne_id%C3%A9ale
- https://fr.wikipedia.org/wiki/Loi_multinomiale
- https://en.wikipedia.org/wiki/Multivariate_normal_distribution
- https://www.youtube.com/channel/UCmpptkXu8iIFe6kfDK5o7VQ
- https://www.caam.rice.edu/~heinken/latex/symbols.pdf
- https://matplotlib.org/stable/index.html

Figure B.1: *Langages et éditeurs utilisés*