INF8775 – Analyse et conception d'algorithmes

Hiver 2021

Laboratoire 1

Travail présenté à :

Mohammed Najib Haouas

1897222 – Alexandre Morinvil

Xxxxxxx – Mohamed Laziz Taoual

Date de remise :

Vendredi 5 mars 2021

# 

# Table des matières

[Table des matières 2](#_Toc64250448)

[Instruction 3](#_Toc64250449)

[Question 1 4](#_Toc64250450)

[Hypothèse 1: 4](#_Toc64250451)

[Hypothèse 2 : 4](#_Toc64250452)

[Hypothèse 3 : 4](#_Toc64250453)

[Question 2 5](#_Toc64250454)

[Hypothèse 1: 5](#_Toc64250455)

[Hypothèse 2 : 7](#_Toc64250456)

[Hypothèse 3 : 9](#_Toc64250457)

[Question 3 12](#_Toc64250458)

[Fonction de génération de messages 12](#_Toc64250459)

[Classe de compression par encodage de Huffman 14](#_Toc64250460)

[Classe de compression par encodage Lz77 15](#_Toc64250461)

[Fonctions utilitaires de gestion de métriques 16](#_Toc64250462)

[Question 4 18](#_Toc64250463)

[Hypothèse 1 : 18](#_Toc64250464)

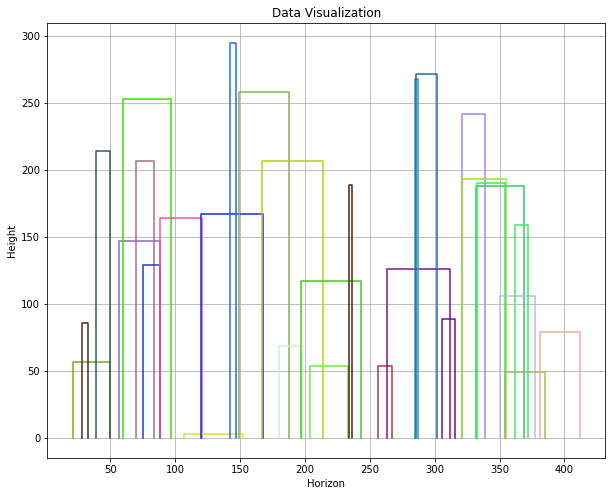
[Hypothèse 2 : 19](#_Toc64250465)

[Hypothèse 3 : 20](#_Toc64250466)

# Instruction

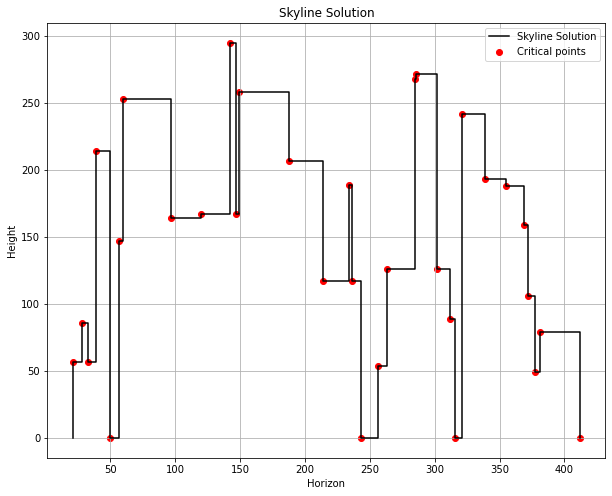
Ce TP vise à se familiariser avec l’analyse empirique et hybride d’algorithmes.

Ainsi, l’objectif de ce TP se d’implémenter deux algorithmes permettant de régler le « Skyline Problem » (Le problème de la ligne d’horizon). Ce problème consister à partir d’un ensemble de données définissant des rectangles juxtaposés et superposés tel qu’affiché dans l’image suivante :



**Figure 1 :** Exemple de visualisation d’un ensemble de données devant être traité

À partir de ces données, un ensemble de points critiques permettant de définir le contour de l’horizon doit être déterminé ce qui permet d’arriver à une représentation analogue à la figure suivante :



**Figure 2 :** Exemple de visualisation des points critiques associés aux données de la figure 1

Un algorithme de résolution par force brute ainsi qu’un algorithme de résolution par utilisation du patron de conception « diviser pour régner » avec seuil de récursivité ajustable seront employés pour résoudre le problème.

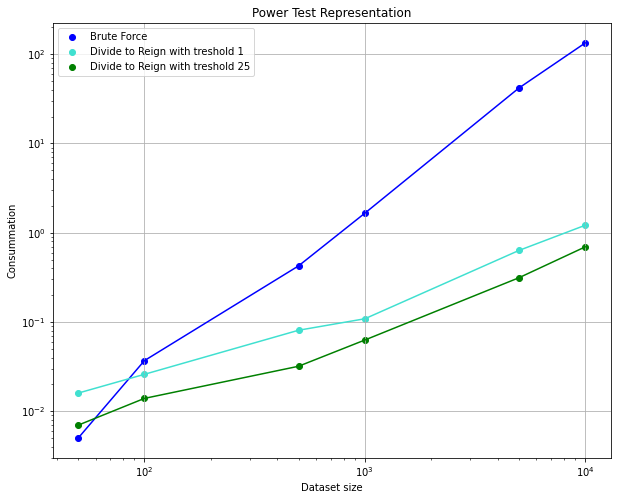
Ensuite, un test de puissance, de rapport et de constante seront utilisés pour analyse l’a complexité asymptotique des algorithmes. Enfin, les résultats obtenus seront analysés.

# Présentation des résultats

Le tableau suivant affiche le temps d’exécution des exemplaires pour chaque taille d’exemplaire.

|  |  |  |  |
| --- | --- | --- | --- |
| Taille de l’exemplaire | Algorithme force brute  [s] | Algorithme diviser pour régner avec seuil de récursivité = 1  [s] | Algorithme diviser pour régner avec seul de récursivité = 25  [s] |
| 50 | 0.008974790573120117 | 0.00898432731628418 | 0.004979610443115234 |
| 100 | 0.03391075134277344 | 0.01894688606262207 | 0.006982564926147461 |
| 500 | 0.3809785842895508 | 0.05385613441467285 | 0.02393627166748047 |
| 1000 | 1.3055057525634766 | 0.11469340324401855 | 0.057845354080200195 |
| 5000 | 38.59204292297363 | 1.2207386493682861 | 0.5275835990905762 |
| 10000 | 163.28127312660217 | 2.219240665435791 | 0.8696675300598145 |
| 50000 | 3614.1043617725372 | 5.6718409061431885 | 5.6718409061431885 |
| 100000 |  |  |  |
| 500000 |  |  |  |

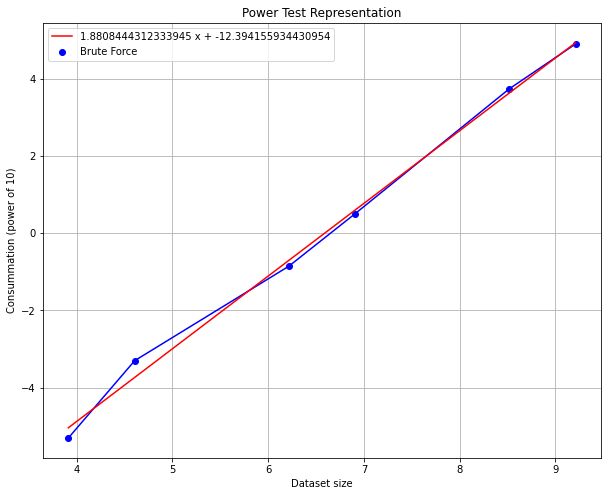
Les points de ce tableau sont affichés dans la figure suivante :



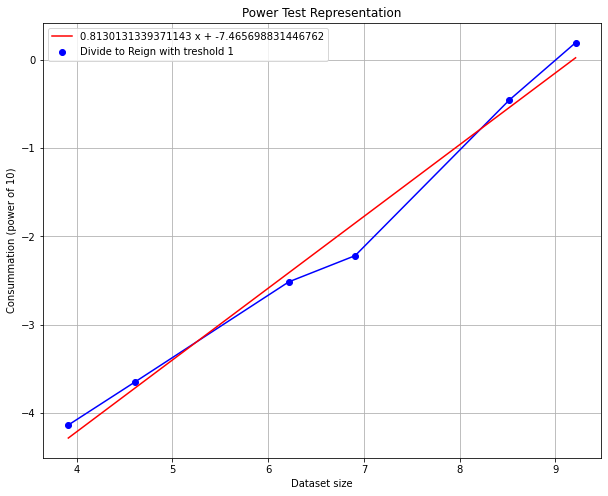
**Figure 3 :** Consommation temporelle en fonction de la taille de l’exemplaire

## Test de puissance :

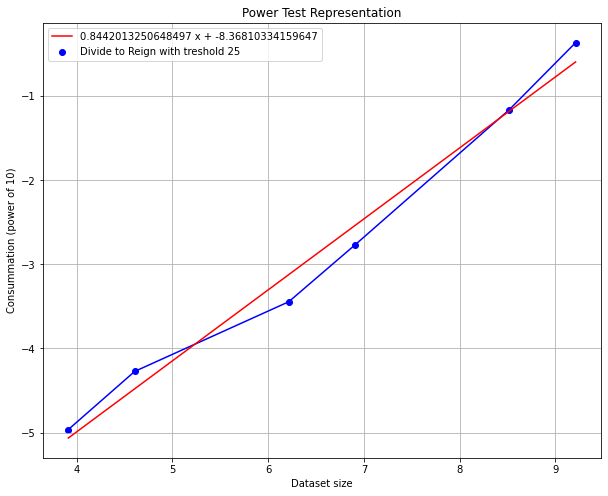
Pour chacun des algorithmes, le test de puissance a été appliqué. Les graphiques obtenus sont affichés ci-bas.



**Figure 4 :** Test de puissance pour l’algorithme de force brute



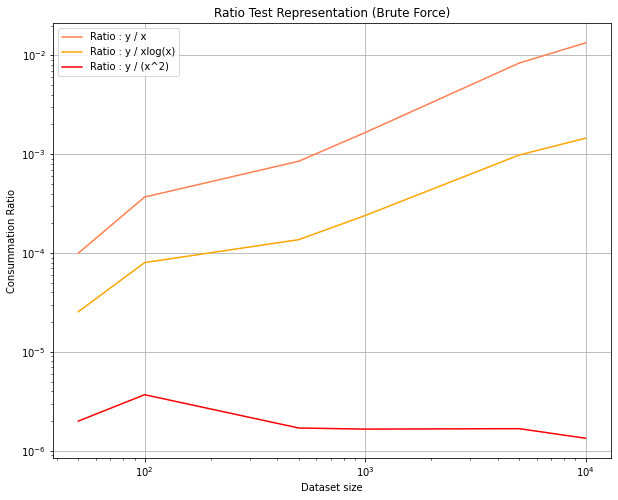
**Figure 5 :** Test de puissance pour l’algorithme diviser pour régner avec seuil de récursivité de 1



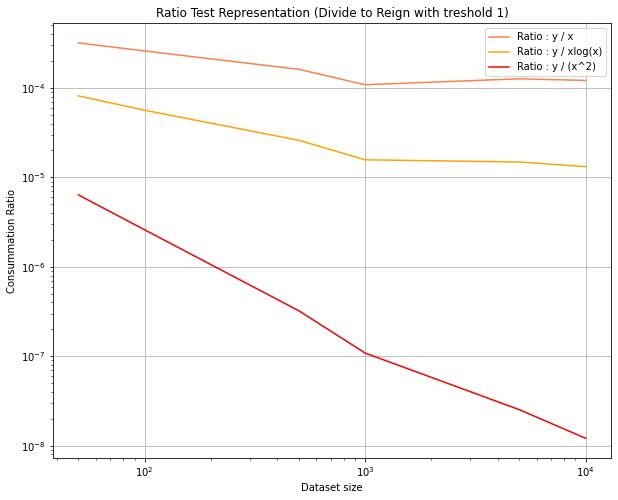
**Figure 6 :** Test de puissance pour l’algorithme diviser pour régner avec seuil de récursivité de 25

## Test du rapport :

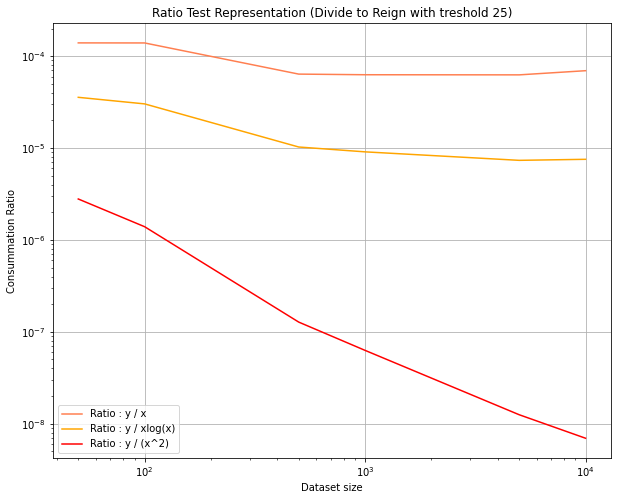
Pour chacun des algorithmes, le test du rapport a été appliqué. Les graphiques obtenus sont affichés ci-bas.



**Figure 7 :** Test du rapport pour l’algorithme de force brute



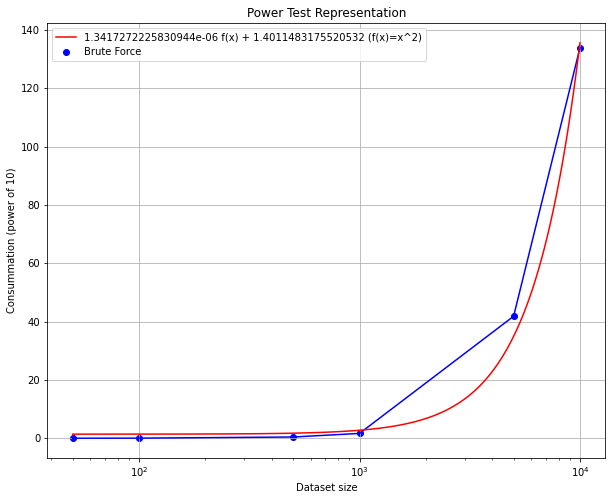
**Figure 8 :** Test du rapport pour l’algorithme diviser pour régner avec seuil de récursivité de 1



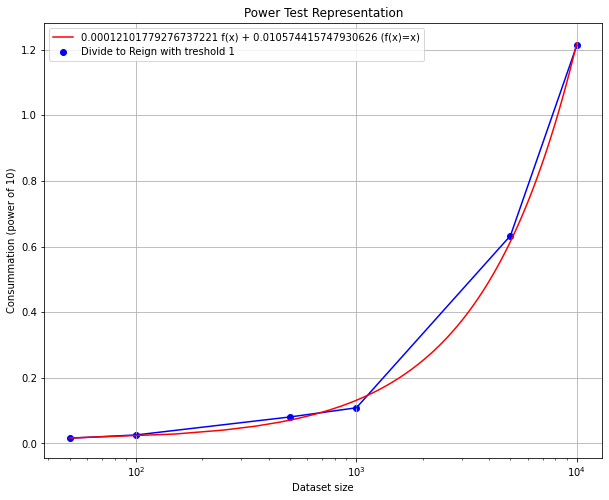
**Figure 9 :** Test de puissance pour l’algorithme diviser pour régner avec seuil de récursivité de 25

## Test des constantes :

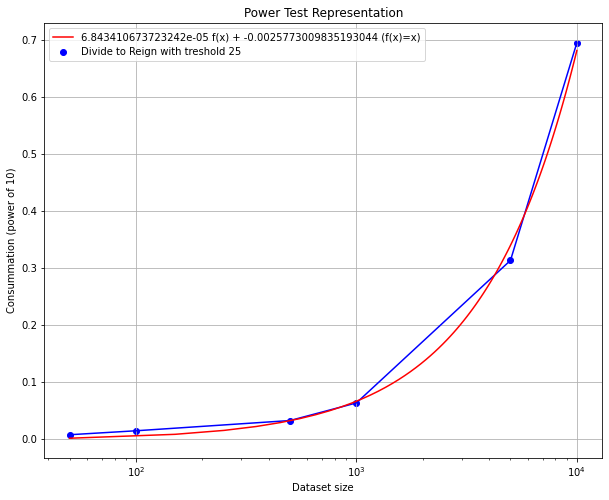
Pour chacun des algorithmes, le test du rapport a été appliqué. Les graphiques obtenus sont affichés ci-bas.



**Figure 10 :** Test des constantes pour l’algorithme de force brute



**Figure 11 :** Test des constantes pour l’algorithme diviser pour régner avec seuil de récursivité de 1



**Figure 12 :** Test de puissance pour l’algorithme diviser pour régner avec seuil de récursivité de 25

# Analyse et discussion

## Déductions réalisées suite au test de puissance.

## Consommation théorique du temps de calcul pour les algorithmes, en notation asymptotique.

Les consommations théoriques du temps de calcul pour chacun des algorithmes, en notation asymptotique, sont les suivantes :

## Déductions réalisées suite au test du rapport.

## Déductions réalisées suite au test de puissance.

## Déductions réalisées suite au test des constantes.

## Observations sur le seuil de récursivité.

## Suite à l’analyse, observations concernant les conditions (taille d’exemplaire ou autre) sous lesquelles chacun de ces algorithmes serait préférable et justification.