école dE TECHNOLOGIE SUPÉRIEURE

UNIVERSITÉ DU QUÉBEC

Rapport de laboratoire 1

PRÉSENTÉ À

SAbeur Lafi

Dans le cadre du cours :

ALGORITHMES

ELE440-01

par

Julien lemay (LEMJ16059303)

Alexandre Lessard (LESA30099400)

montréal, le 2 octobre 2015

cid:image002.jpg@01CCE021.010CA410 Julien Lemay et Alexandre Lessard, 2015

Sommaire

[Introduction 1](#_Toc431509799)

[1. Tri par insertion 2](#_Toc431509800)

[1.1. Algorithme 2](#_Toc431509801)

[1.2. Analyse théorique 2](#_Toc431509802)

[1.3. Analyse expérimentale 2](#_Toc431509803)

[2. Tri par fusion 3](#_Toc431509804)

[2.1. Algorithme 3](#_Toc431509805)

[2.2. Analyse théorique 3](#_Toc431509806)

[2.3. Analyse expérimentale 3](#_Toc431509807)

[3. Tri par pigeonnier 4](#_Toc431509808)

[3.1. Algorithme 4](#_Toc431509809)

[3.2. Analyse théorique 4](#_Toc431509810)

[3.3. Analyse expérimentale 4](#_Toc431509811)

[4. Tri par tas 5](#_Toc431509812)

[4.1. Algorithme 5](#_Toc431509813)

[4.2. Analyse théorique 5](#_Toc431509814)

[4.3. Analyse expérimentale 5](#_Toc431509815)

[5. Tri rapide 6](#_Toc431509816)

[5.1. Algorithme 6](#_Toc431509817)

[5.2. Analyse théorique 6](#_Toc431509818)

[5.3. Analyse expérimentale 6](#_Toc431509819)

[6. Tri par base 7](#_Toc431509820)

[6.1. Algorithme 7](#_Toc431509821)

[6.2. Analyse théorique 7](#_Toc431509822)

[6.3. Analyse expérimentale 7](#_Toc431509823)

[Conclusion 8](#_Toc431509824)

# Introduction

Le but de ce laboratoire est d’étudier le comportement et la performance de plusieurs algorithmes de tri selon différentes configurations~~.~~

Les objectifs de ce laboratoire sont l’implémentation des différents algorithmes en langage de programmation C, C++ ou Java ainsi que d’utiliser l’analyse asymptomatique afin d’analyser les performances des algorithmes de tri.

Ce rapport comporte trois grandes parties pour chacun des 6 algorithmes. La première partie contient les pseudo-codes qui forment le squelette de nos fonctions de tri. La seconde contient l’analyse théorique où l’on discute de l’efficacité des algorithmes se basant sur leurs formules asymptotiques et baromètres. La troisième section contient l’analyse expérimentale des 6 méthodes de tri c'est-à-dire une description de l’algorithme de test, les résultats obtenus par les 6 fonctions et les conclusions qu’on peut en tirer.

# Tri par insertion

## Algorithme

Voici l’algorithme de tri par insertion tiré de la page Wikipedia de cet algorithme1:

**procédure** tri\_insertion(tableau T, entier n)

**pour** i de 1 à n-1

x ← T[i]

j ← i

**tant que** j > 0 et T[j - 1] > x

T[j] ← T[j - 1]

j ← j - 1

**fin tant que**

T[j] ← x

**fin pour**

**fin procédure**

## Analyse théorique

Nos baromètres sont : }

Dans le meilleurs des cas, le tableaux est en ordre et on ne rentre qu’une fois dans la boucle « tant que » de la ligne 5.

Dans le pire des cas, le tableaux est en désordre total et on rentre N fois dans la boucle « tant que » de la ligne 5.

## Analyse expérimentale

# Tri par fusion

## Algorithme

Voici l’algorithme de tri par fusion tiré de la page Wikipedia de cet algorithme2:

**procédure** tri\_fusion(tableau t)

n = longueur(t)

**si** n > 1 **alors**

u = tri\_fusion(t[1], …, t[n / 2]) // première moitié

v = tri\_fusion(t[n / 2 + 1], …, t[n]) // deuxième moitié

a = 1 // pointeur pour parcourir les éléments de u

b = 1 // pointeur pour parcourir les éléments de v

**pour** i **allant de** 1 à n **faire**

**si** (a <= longueur(u)) // si tous les éléments de u n'ont pas été parcourus

et (b > longueur(v) // et si tous les éléments de v ont été parcourus

ou u[a] ≤ v[b]) alors // ou si l'élément courant de u est plus petit

t[i] = u[a] // on choisit l'élément courant de u comme élément suivant de t

a = a + 1 // on avance dans u

**sinon**

t[i] = v[b] // on choisit l'élément courant de v comme élément suivant de t

b = b + 1 // on avance dans v

**fin si**

**fin pour**

**fin si**

**renvoyer** t

**fin procédure**

## Analyse théorique

## Analyse expérimentale

# Tri par pigeonnier

## Algorithme

**procédure** tri\_pigeonnier(Tableau de données, Nombre de données, Valeur maximum)

**Pour** la valeur maximal d’un nombre dans le tableau

Initialiser cette valeur à zéro.

**Pour** le nombre de données dans le tableau

Placer la valeur dans son pigeonnier respectif

**Pour** tous les chiffres du pigeonnier

**Tant** qu’il y a des chiffres dans le pigeonnnier

Placer la valeur directement après la dernière valeur placé dans le tableau trié.

Incrémenter de 1 le chiffre à remplacer dans le tableau à retourner.

**fin procédure**

## Analyse théorique

## Analyse expérimentale

# Tri par tas

## Algorithme

## Analyse théorique

## Analyse expérimentale

# Tri rapide

## Algorithme

Voici l’algorithme de tri par rapide tiré de la page Wikipedia de cet algorithme5:

**partitionner**(tableau T, entier premier, entier dernier, pivot)

{

**échanger** T[pivot] et T[dernier]

j :=premier

**pour** i de premier à dernier - 1

{

si T[i] <= T[dernier] alors

échanger T[i] et T[j]

j := j + 1

}

échanger T[dernier] et T[j]

**renvoyer** j

}

**tri\_rapide**(tableau T, entier premier, entier dernier)

**début**

**si** premier < dernier **alors**

pivot := choix\_pivot(T, premier, dernier)

pivot := partitionner(T, premier, dernier, pivot)

tri\_rapide(T, premier, pivot-1)

tri\_rapide(T, pivot+1, dernier)

**fin si**

**fin**

## Analyse théorique

## Analyse expérimentale

# Tri par base

## Algorithme

## Analyse théorique

## Analyse expérimentale

# Conclusion

# Annexe

# Références

1 :Wikipedia. Tri par insertion,[En ligne], https://fr.wikipedia.org/wiki/Tri\_par\_insertion (Consulté le 1er octobre 2015).

2 : Wikipedia. Tri fusion,[En ligne], https://fr.wikipedia.org/wiki/Tri\_fusion  
 (Consulté le 1er octobre 2015).

5 :Wikipedia. Tri rapide,[En ligne], https://fr.wikipedia.org/wiki/Tri\_rapide   
(Consulté le 1er octobre 2015).