**Laborationsrapport  
Algoritmer och datastrukturer I , 7,5 HP B  
HT2015**

**Laboration nr: 4**

**Exekveringstid for sorteringsalgoritmer**

av

Viktor Hanstorp (19940413)

**Institutionen för matematik, natur- och datavetenskap**

**Högskolan i Gävle**

S-801 76 Gävle, Sweden

Email:

*ndi14vhp@student.hig.se*

**Innehåll**

1 Inledning 2

2 Förutsättningar och krav 2

2.1 Uppgifter 2

2.2 Förutsättningar 2

3 Resultat 2

4 Diskussion 3

4.1 Uppgift 1 3

4.1.1 Fråga 1 3

4.1.2 Fråga 2 3

4.1.3 Fråga 3 3

4.2 Uppgift 2 4

4.2.1 Insertionsort 4

4.2.2 Quicksort 4

4.2.3 Bubblesort 4

4.2.4 Mergesort 4

5 Sammanfattning 4

6 Referenser 5

# Inledning

Laborationen går ut på att lära sig mer om sorteringsalgoritmer, användandet av empiriska studier för att analysera dem, samt dess ordonalitet.

# Förutsättningar och krav

## Uppgifter

* Lista ut vilken strategi som används av den givna implementationen av Quicksort för att välja pivotelement
* Motivera varför vissa tider avviker i det givna programmet
* Lista ut i vilken ordning som sorteringsalgoritmerna i det givna programmet körs.
* Visa antalet jämförelser utförda i sorteringsalgoritmerna med hjälp av artimetiska serier

## Förutsättningar

De sorteringsalgoritmer som används är:

* Bubblesort
* Insertionsort
* Mergesort
* Quicksort

|  |  |  |  |
| --- | --- | --- | --- |
| **Algoritm** | **Bäst** | **Medel** | **Värst** |
| Bubblesort |  |  |  |
| Insertionsort |  |  |  |
| Mergesort |  |  |  |
| Quicksort |  |  |  |

Tabell - Tid för olika sorteringsalgoritmer

Enligt wikipedia [1] så är tidsåtgången för sorteringsalgoritmerna enligt Tabell 1

Artimetiska summan:

# Resultat

För framtagning av resultat för körningar av det givna programmet skapades ett program för att räkna ut medelvärdet för ett antal körningar (”AOD\_Lab4\_Runner”).

Programmet startar ”AOD\_Lav4.exe” och extraherar tiderna från dess standard-output. Medelvärdet beräknas sedan.

Resultat kan ses i bilaga 1.

# Diskussion

## Uppgift 1

### Fråga 1

Det första värdet väljs som pivotelement för denna implementation av Quicksort.

### Fråga 2

**Metod 1**

”b.txt” och ”d.txt” ger lägre värden än förväntat.

Detta betyder att sorterade (ej omvänt) datamängder är optimalt.

**Metod 2**

”a.txt”, ”b.txt” och ”d.txt” avviker med att vara rejält mycket större än de borde.

Detta betyder att sorterade (även omvänt) datamängder ger dåligt resultat.

**Metod 3**

”b.txt” och ”d.txt” ger typ lägre värden än förväntat.

Detta betyder att sorterade (ej omvänt) datamänger är optimalt.

**Metod 4**

Alla värden ser jämna ut

### Fråga 3

*Mergesort* är den enda sorteringsalgoritm där alla fall är lika. Detta kan endast observeras i metod 4.

”Divide and conquer” metoder ger de bästa tiderna för jämförelsebaserade sorteringar, , detta kan ses i metod 2 och metod 4. Då metod 4 är mergesort så är metod 2 *Quicksort*.

*Bubblesort* är optimal då datamängden är sorterad, då det inte behöver göras några byten. Detta kan observeras i metod 3 (~ 0 för sorterat).

Uteslutningsmetoden säger att metod 1 är *Insertionsort*. Detta kan även bekräftas då sortering av sorterad datamängd tar längre tid än metod 3.

**Metod 1 – Insertionsort**

**Metod 2 – Quicksort**

**Metod 3 – Bubblesort**

**Metod 4 – Mergesort**

## Uppgift 2

Uträkning av tidsåtgång för jämnföring i värsta fallet.

Gällande givna algoritmer.

### Insertionsort

### Quicksort

### Bubblesort

### Mergesort

# Sammanfattning

Resultatet av uträkningarna i Uppgift 2 stämmer med teorin [1], vilket betyder att det kan vara korrekt.

# Referenser

|  |  |
| --- | --- |
| [1] | Wikipedia, "Sorting algorithm --- Wikipedia{,} The Free Encyclopedia," 2015. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Sorting\_algorithm&oldid=694927729. [Accessed 20 12 2015]. |