

# **Report**

## **Assignment 3 : Advance Sorting**

จัดทำโดย

นายสรธร แก้วโชติช่วงกุล 63070501067 CPEREGULAR

รายงานนี้เป็นส่วนหนึ่งของวิชา CPE111

Programming With Data Structures

King Mongkut's University of Technology Thonburi

## ● สิ่งที่ทำใน Assignment

1. สร้าง class DataArray และฟังก์ชันในการทำงานเกี่ยวกับ data หลายฟังก์ชันเพื่อจัดการกับข้อมูลได้แก่ ฟังก์ชันในการอ่านไฟล์, ฟังก์ชันในการเพิ่มข้อมูลเข้า Array
2. สร้าง class CSVnode เพื่อมาเก็บข้อมูลของข้อมูลแต่ละบรรทัดที่ประกอบด้วยข้อมูลอยู่ 4 필ด์
3. สร้างตัวแปร Array เพื่อมาเก็บข้อมูลที่อ่านได้จากไฟล์
4. สร้างฟังก์ชันของ sort ที่ต้องการทดสอบทั้งจากน้อยไปมากและเรียงลำดับ String ตามรหัส Ascii ได้แก่ Library sort, merge sort, Quick sort
5. สร้างฟังก์ชันในการสลับข้อมูล
6. สร้างฟังก์ชันในการแสดงผลตามจำนวนที่ต้องการทดสอบ
7. จับเวลาการทำงานของ sort แต่ละอันและนำมาแสดงผลเป็นตาราง

- Source code

- Class ที่ใช้ Run

```
1 import java.util.*;
2
3 public class Ass3 {
4
5     public static void main(String[] args) {
6         int max = 100010;
7         double timenumlibrary, timestrlibrary, timenummerge, timestrmerge, timenumquick
8         , timestrquick;
9         long starttime, stoptime;
10        dataArray test = new dataArray(100010);
11        dataArray temp = new dataArray(100010);
12        if(test.Load_DataFile("C:\\Users\\LENOVO\\Downloads\\testAss3.csv") > 0) {
13            System.out.println("System Can Read File = "+test.count); // แสดงจำนวนข้อมูลที่อ่านได้
14            //No sort
15            System.out.println("No sorting.....");
16            test.print_test();
17
18            // Arrays sort num
19
20            temp.data = Arrays.copyOf(test.data, test.count); // copy ข้อมูลเพื่อใช้ในการเรียงในแบบต่างๆ
21            temp.count = test.count;
22            System.out.println("Run Arrays sort number.....");
23            starttime = System.nanoTime();
24            Arrays.sort(temp.data, new Arraynum());
25            stoptime = System.nanoTime();
26            timenumlibrary = (stoptime-starttime) / 1E6;
27            temp.print_test();
28
29            // Arrays sort string
30
31            temp.data = Arrays.copyOf(test.data, test.count); // copy ข้อมูลเพื่อใช้ในการเรียงในแบบต่างๆ
32            temp.count = test.count;
33            System.out.println("Run Arrays sort Strings.....");
34            starttime = System.nanoTime();
35            Arrays.sort(temp.data, new ArrayStr2());
36            stoptime = System.nanoTime();
37            timestrlibrary = (stoptime-starttime) / 1E6;
38            temp.print_test();
39
40            //Merge Sort num
41
42            temp.data = Arrays.copyOf(test.data, test.count); // copy ข้อมูลเพื่อใช้ในการเรียงในแบบต่างๆ
43            temp.count = test.count;
44            System.out.println("Run Merge sort number.....");
45            starttime = System.nanoTime();
46            temp.mergeSort(0, temp.count-1);
47            stoptime = System.nanoTime();
48            timenummerge = (stoptime-starttime) / 1E6;
49            temp.print_test();
50
51            // Merge Sort String
52
53            temp.data = Arrays.copyOf(test.data, test.count); // copy ข้อมูลเพื่อใช้ในการเรียงในแบบต่างๆ
54            temp.count = test.count;
55            System.out.println("Run Merge sort String.....");
56            starttime = System.nanoTime();
57            temp.mergeSortString(0, temp.count-1);
58            stoptime = System.nanoTime();
59            timestrmerge = (stoptime-starttime) / 1E6;
60            temp.print_test();
61
62            // Quicksort num
63
64            temp.data = Arrays.copyOf(test.data, test.count); // copy ข้อมูลเพื่อใช้ในการเรียงในแบบต่างๆ
```

```

65 temp.count = test.count;
66 System.out.println("Run Quick sort Number.....");
67 starttime = System.nanoTime();
68 temp.QuickSortNum(0,temp.count-1);
69 stoptime = System.nanoTime();
70 timenumquick = (stoptime-starttime) / 1E6;
71 temp.print_test();
72
73 //Quicksort string
74
75 temp.data = Arrays.copyOf(test.data,test.count); // copy ข้อมูลเพื่อใช้ในการจัดเรียงแบบต่างๆ
76 temp.count = test.count;
77 System.out.println("Run Quick sort String.....");
78 starttime = System.nanoTime();
79 temp.QuickSortStr(0,temp.count-1);
80 stoptime = System.nanoTime();
81 timestrquick = (stoptime-starttime) / 1E6;
82 temp.print_test();
83
84 System.out.println("End of testing."); // คำนวณได้ไม่ตรงกับจำนวนของการทดสอบเพราะการรันซ้ำในแบบต่างของแต่ละอันเป็นรูปแบบเฉพาะ
85 System.out.printf("\t\t|-----+\n\n");
86 System.out.printf("\t\t|          Sort          |   time Number (ms)   | time String (ms) |\n\n");
87 System.out.printf("\t\t|-----+\n\n");
88 System.out.printf("\t\t| Library sort      |    %10.6f|    %10.6f ms|\n",timenumlibrary,timestrlibrary);
89 System.out.printf("\t\t| Merge sort       |    %10.6f|    %10.6f ms|\n",timenummerge,timstrmerge);
90 System.out.printf("\t\t| Quick sort       |    %10.6f|    %10.6f ms|\n",timenumquick,timstrquick);
91 System.out.printf("\t\t|-----+\n\n");
92 }
93 else
94     System.out.printf("Can't Read File");
95 System.out.printf("This Program Written By Sorathorn Kaewchotchuangkul 63070501067 CPeregular");
96

```

- Class ที่ใช้เก็บข้อมูลของ Array

```

2 public class CSVnode {
3     int id;
4     long num;
5     String str1, str2;
6     public CSVnode(){
7
8     }
9     public CSVnode(int a,long b,String c,String d){
10         id = a;
11         num = b;
12         str1 = c;
13         str2 = d;
14     }
15     public void print_data(){
16         System.out.printf("%6d %d %s %s\n",id,num,str1,str2);
17     }
18 }
19

```

```

16•     public int compareTo(CSVnode x) {
17         if(num - x.num > 0)
18             return 1;
19         else if (num - x.num == 0)
20             return 0;
21         else
22             return -1;
23     }
24•     public void print_data(){ // method ในการแสดงผลข้อมูลของ data แต่ละตัว
25         System.out.printf("%6d %d %s %s\n",id,num,str1,str2);
26     }
27 }
28 class Arraynum implements Comparator <CSVnode>{
29•     public int compare(CSVnode x,CSVnode y) {
30         if(x.num == y.num)
31             return 0;
32         else if (x.num > y.num)
33             return 1;
34         else
35             return -1;
36     }
37 }
38 class ArrayStr2 implements Comparator <CSVnode>{
39•     public int compare(CSVnode x,CSVnode y) {
40         return (int)x.str2.compareToIgnoreCase(y.str2);
41     }
42 }

```

## - Class ที่จัดการกับ Data

```
1 import java.io.File;
4
5 public class DataArray {
6     CSVnode [] data;
7     int count;
8     public DataArray(int max){
9         data = new CSVnode[max];
10        count = 0;
11    }
12    public void add_data(int a,long b,String c,String d){
13        CSVnode x = new CSVnode(a,b,c,d);
14        data[count++] = x;
15    }
16    public void print_test(int start,int stop) {
17        int i;
18        for(i=start;i<stop;i++)
19            data[i].print_data();
20    }
21    public void swapdata(int i,int j){
22        CSVnode x = new CSVnode();
23        x = data[j];
24        data[j] = data[i];
25        data[i] = x;
26    }
```

## ➤ method Quicksort สำหรับสตริง

```
27 public void QuicksortStr(int first, int last){
28     int i=first, j=last;
29     if (first < last){
30         do
31         {
32             while((data[i].str2.compareToIgnoreCase(data[j].str2) <= 0) && (i < j))
33                 j--;
34             if (data[i].str2.compareToIgnoreCase(data[j].str2) > 0){
35                 swapdata(i,j);
36                 i++;
37             }
38             while((data[i].str2.compareToIgnoreCase(data[j].str2) <= 0) && (i < j))
39                 i++;
40             if (data[i].str2.compareToIgnoreCase(data[j].str2) > 0){
41                 swapdata(i,j);
42                 j--;
43             }
44         } while(i<j);
45         if (first < j-1)
46             QuicksortStr(first, j-1);
47         if (i+1 < last)
48             QuicksortStr(i+1, last);
49     }
```

## ➤ Quick sort สำหรับ Number

```
51• public void QuicksortNum(int first, int last){
52     int i=first, j=last;
53     if (first < last){
54         do
55         {
56             while((data[i].num <= data[j].num) && (i < j))
57                 j--;
58             if (data[i].num > data[j].num){
59                 swapdata(i,j);
60                 i++;
61             }
62             while((data[i].num <= data[j].num) && (i < j)){
63                 i++;
64             }
65             if (data[i].num > data[j].num){
66                 swapdata(i,j);
67                 j--;
68             }
69         } while(i<j);
70         if (first < j-1)
71             QuicksortNum(first, j-1);
72         if (i+1 < last)
73             QuicksortNum(i+1, last);
74     }
75 }
```

## ➤ Merge Sort สำหรับ String

```
97• public void mergeDataString(int first,int last){
98     int mid, i, i1, i2;
99     CSVnode[] temp = new CSVnode[last - first + 1];
100     mid = (first + last) / 2;
101     i1 = first;
102     i2 = mid+1;
103     for (i=0;i<=last-first;i++){
104         temp[i] = new CSVnode();
105         if (i1<=mid && i2<=last){
106             if ((data[i1].str2.compareToIgnoreCase(data[i2].str2) < 0))
107                 temp[i] = data[i1++];
108             else
109                 temp[i] = data[i2++];
110         }
111         else if (i1>mid)
112             temp[i] = data[i2++];
113         else if (i2>last)
114             temp[i] = data[i1++];
115     }
116     for (i=0;i<=last-first;i++)
117         data[first+i] = temp[i];
118 }
```

➤ Merge sort สำหรับ Number

```
75• public void mergeDatanum(int first,int last){
76     int mid, i, i1, i2;
119• public void mergeSort(int first, int last){
120     int mid;
121     if (first < last){
122         mid = (first+last)/2;
123         mergeSort(first, mid);
124         mergeSort(mid+1,last);
125         mergeDatanum(first,last);
126     }
127 }
128• public void mergeSortString(int first, int last){
129     int mid;
130     if (first < last){
131         mid = (first+last)/2;
132         mergeSortString(first, mid);
133         mergeSortString(mid+1,last);
134         mergeDataString(first,last);
135     }
136 }
96 }
```

**Note!** อธิบายโค้ด comment อยู่ในตัวโค้ด



- Test case & อธิบาย

Testcase : จากน้อยไปมากของทุก sort ที่เรียงจากตัวเลขและเรียงข้อความจาก A-Z

[illegible]

**กรอบสีแดง** : เป็น Test case แบบยังไม่ได้จัดเรียงข้อมูลอะไรเลย

**กรอบสี่เขียว** : เป็น Test case ของการเรียงแบบ Library หรือ Arrays sort ของ Java โดยจะมีทั้งเรียงแบบตัวเลขและเรียงจากตัวอักษรตามลำดับและได้ผลตามที่อาจารย์ให้มา

**กรอบสีม่วง :** เป็น Test case ของการเรียงแบบ Merge sort โดยจะมีทั้งเรียงแบบ  
ตัวเลขและเรียงจากตัวอักษรตามลำดับและได้ผลตามที่อาจารย์ให้มา

**กรอบสี่เหลี่ยม :** เป็น Test case ของการเรียงแบบ Quick sort โดยจะมีทั้งเรียงแบบ  
ตัวเลขและเรียงจากตัวอักษรตามลำดับและได้ผลตามที่อาจารย์ให้มา

Test case : ผลลัพธ์ของเวลาที่ใช้ในการเรียงข้อมูลที่ได้

Sort	time Number (ms)	time String (ms)
Library sort	63.298900	236.136900 ms
Merge sort	77.217600	104.712800 ms
Quick sort	50.348700	150.150700 ms

=== สรุปผล ===

Library sort : เวลาที่ใช้จากการเรียงลำดับแบบ Number และ String มีค่าประมาณ 63 และ 236ms. ตามลำดับ

Merge sort : เวลาที่ใช้จากการเรียงลำดับแบบ Number และ String มีค่าประมาณ 77 และ 104ms. ตามลำดับ

Quick sort : เวลาที่ใช้จากการเรียงลำดับแบบ Number และ String มีค่าประมาณ 50 และ 150 ms. ตามลำดับ

เวลาที่ใช้ ในการเรียงลำดับแบบต่างๆจะสรุปได้ดังนี้

Number (จากน้อยไปมาก) : Quick sort < Library sort < Merge sort

String (จาก A-Z ตามรหัส Ascii) : Merge sort < Quick sort < Library sort

สรุป : การเรียงจากน้อยไปมากแบบ Number ➡ Quick sort เร็วที่สุด

การเรียงจากน้อยไปมากตามรหัส Ascii String ➡ Merge sort เร็วที่สุด

- สรุปความเข้าใจของตนเอง

ใน Assignment นี้เราต้องนำข้อมูลที่ได้จากการอ่านจากไฟล์มาทดสอบเรียงลำดับ 2 แบบได้แก่แบบ Number และ String และหาเวลาในการเรียงลำดับแบบต่างๆโดยมีวิธีการเรียงลำดับอยู่ 3 วิธีคือ Library sort, Merge sort และ Quick sort และนำมาแสดงผลดูว่าการเรียงลำดับแบบไหนใช้เวลา น้อยที่สุดและสรุปผล

- ผลการประเมินตนเอง

ให้ตนเองอยู่ที่ระดับ 80 เพราะสามารถทำโจทย์ได้ด้วยตัวเองและมีปัญหาเล็กน้อยในส่วนของการ สร้าง method ของแต่ละ sort และยังไม่มั่นใจ 100 % ในการทำโจทย์ประยุกต์จากโจทย์นี้

