

Report : Assignment 2

Made By : Sorathorn Kaewchotchuankul 63070501067 CPE REGULAR

● สิ่งที่ทำใน Assignment

1. สร้าง class DataArray และฟังก์ชันในการทำงานเกี่ยวกับ dataหลายฟังก์ชันเพื่อจัดการกับข้อมูลได้แก่ ฟังก์ชันในการอ่านไฟล์, ฟังก์ชันในการเพิ่มข้อมูลเข้า Array
2. สร้าง class CSVnode เพื่อมาเก็บข้อมูลของข้อมูลแต่ละบรรทัดที่ประกอบด้วยข้อมูลอยู่ 4 필ด์
3. สร้างตัวแปร Array เพื่อมาเก็บข้อมูลที่อ่านได้จากไฟล์
4. สร้างฟังก์ชันของ sort ที่ต้องการทดสอบทั้งจากน้อยไปมากและมากไปน้อย
5. สร้างฟังก์ชันในการสลับข้อมูล
6. สร้างฟังก์ชันในการแสดงผลตามจำนวนที่ต้องการทดสอบ
7. จับเวลาการทำงานของ sort แต่ละอันและนำมาแสดงผลเป็นตาราง

- 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
```

- Class ที่จัดการกับ Data

```
1 import java.io.File;
2
3
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    }
27 }
```

```

27* public void Insertion_sort(int start,int stop,int order) {
28     int i,j;
29     CSVnode x = new CSVnode();
30     if(order == 0) {
31         for(i=start+1;i<=stop;i++) {
32             x = data[i];
33             for(j=i;j>start && (x.num < data[j-1].num);j--)
34                 data[j] = data[j-1];
35             data[j]= x;
36         }
37     }
38     else {
39         for(i=start+1;i<=stop;i++) {
40             x = data[i];
41             for(j=i;j>start && (x.num > data[j-1].num);j--) // j and i is the same but use to compare value
42                 data[j] = data[j-1];
43             data[j]= x;
44         }
45     }
46 }
47
48 }
49● public void Bubble_sort(int start,int stop,int order) {
50     int i,j;
51     boolean doswap = true;
52     if(order == 0) {
53         for(i=start;i<stop && doswap ;i++) {
54             doswap = false;
55             for(j=stop;j>i;j--) {
56                 if(data[j].num < data[j-1].num) {
57                     swapdata(j,j-1);
58                     doswap = true;
59                 }
60             }
61         }
62     }
63     else {
64         for(i=start;i<stop && doswap ;i++) {
65             doswap = false;
66             for(j=stop;j>i;j--) {
67                 if(data[j].num > data[j-1].num) {
68                     swapdata(j,j-1);
69                     doswap = true;
70                 }
71             }
72         }
73     }
74 }

```

```

75• public void selection_sort(int start,int stop,int order) {
76     int i,j,min;
77     if(order == 0) {
78         for(i=start;i<stop;i++) {
79             min = i;
80             for(j=i+1;j<=stop-1;j++) {
81                 if(data[j].num < data[min].num)
82                     min = j;
83             }
84             swapdata(min,i);
85         }
86     }
87     else {
88         for(i=start;i<stop;i++) {
89             min = i;
90             for(j=i+1;j<=stop-1;j++) {
91                 if(data[j].num > data[min].num)
92                     min = j;
93             }
94             swapdata(min,i);
95         }
96     }
97 }
98• public void scansort(int start,int stop,int order) {
99     int i,j;
100    if(order == 0) {
101        for(i=start;i<stop;i++)
102            for(j=i+1;j<=stop;j++)
103                if(data[j].num < data[i].num)
104                    swapdata(i,j);
105    }
106    else {
107        for(i=start;i<stop;i++)
108            for(j=i+1;j<=stop;j++)
109                if(data[j].num > data[i].num)
110                    swapdata(i,j);
111    }
112 }
113• public int Load_DataFile(String filename) {
114     int count2 = 0, a;
115     long b;
116     String c,d;
117     try {
118         File fr = new File(filename);
119         var read = new Scanner(fr);
120         read.useDelimiter(",|\\n");
121         while(read.hasNextInt()) {
122             a = read.nextInt();
123             b = read.nextLong();
124             c = read.next();
125             d = read.nextLine();
126             add_data(a,b,c,d);
127             count2++;
128         }
129     }
130     catch(FileNotFoundException e){
131         System.out.println("Error ! can't read File "+filename);
132         return 0;
133     }
134 }
135 return count2;
136 }
137 }
138

```

Note! อธิบายโค้ด comment อยู่ในตัวโค้ด

Test case : จากมากไปน้อยของทุก sort

```
system compile file 180000
Scansort testing time.....

====++ Scan Sort (Descending) +++====
9614 999935638771 PKVYEACGIRDDIHIVSNQSTUDIINGBYZMVRFYAQWAXAYLZZDIGNMJN ,ZFBVYRDMWIEKAKSCVTSUPJSTQLAYSENVPAVHHTSXHF COHPHPQFD
3351 999923652622 YNJEGMKMIXLQMTAXIFNHDZZBNFEHSJFRJPIINSTYQSVWCEFNZGYGL ,LBXLVPMDVNLSCKKMYYDAOSADKXZBAPTZZXCLEPVVRNAFEZJNDJWH
5837 999917776745 EZJEBKRJECUONVUNNHTUQOMRKUNITKCECFBJSVXXJFDXOV CSTIBW ,WREWLUSBZZLCFWAKJBHORDHDGKQAIWDRWEXVZADKBVXDIVFKNVPC
9040 999751427723 IPXGNLCSWOSUBIOZCHVQKORTLZZYCYLQXEVZYGVBFXLUISFSPDBW ,JUKSZWKCDZDPJNNRBDHCTQKUOMVKGWNLUPQZSLHNCB EYOWVQMQC
6760 999644798594 HZXYQOKKYVSFYWRDGVQESZSLBDSBAMZVQBHGCBJVLHXBEUTQUTLU ,GDWSLQHIAYOYSCGDYUWZQSICFZCKGIDOSRZKEMHEIGXXTALMFFYK
3587 999637438235 FEAZIBNEGHNHUNTHQJMOZGQQSNVJJIWYACOHCCIWLEIENICHDKTHBU ,BIVXBPFXAEXDBZZXKWJNQXLDZIFTHLKFIZNHEIDMAEJQZCXVET
5614 999572594287 LXXKMUCCNZRDIFHVITYIBBTUFQUMHPRZWPQAZGKMAGBBWYWKIKNYP ,KXINQMPRNIE LUPZMHBDXYVBRGAKQYWBPTLKPWUUCFZPPVWVRFZJXJ
555 999566631984 QHQNNDALDOCIJCEVMBYFYSVKCSFNUPNGICYKKEWSPCGQHKNXYG ,XGCVTTZVTFLQVCWKMMAMZIWJPCWHDHFXFRXVUTBGYWMANKDIMMPDL
6643 999527766159 MCDUWUUHVGEENXNHJJVFXTMDGQXCYTIVFGCMHCOREYBTHNNCPR ,XKNWIXJOYYARJFTNKKZUCHTQDNXTYFNDOXAYUCZKJFSKHSSV00UB
2935 999463416692 OKVIBWGNFQSSAEHODDKHRYMXLLSLCFINEVRZIDGLWKKWXYTJZJIZ ,MMLAWCNMAMVKUOWVEMTHCOIRFAPFGOBOWMYZTTHVLOPEDVSAUC
End of Scansort testing.
Selectionsort testing time.....

====++ Selection Sort (Descending) +++====
9614 999935638771 PKVYEACGIRDDIHIVSNQSTUDIINGBYZMVRFYAQWAXAYLZZDIGNMJN ,ZFBVYRDMWIEKAKSCVTSUPJSTQLAYSENVPAVHHTSXHF COHPHPQFD
3351 999923652622 YNJEGMKMIXLQMTAXIFNHDZZBNFEHSJFRJPIINSTYQSVWCEFNZGYGL ,LBXLVPMDVNLSCKKMYYDAOSADKXZBAPTZZXCLEPVVRNAFEZJNDJWH
5837 999917776745 EZJEBKRJECUONVUNNHTUQOMRKUNITKCECFBJSVXXJFDXOV CSTIBW ,WREWLUSBZZLCFWAKJBHORDHDGKQAIWDRWEXVZADKBVXDIVFKNVPC
9040 999751427723 IPXGNLCSWOSUBIOZCHVQKORTLZZYCYLQXEVZYGVBFXLUISFSPDBW ,JUKSZWKCDZDPJNNRBDHCTQKUOMVKGWNLUPQZSLHNCB EYOWVQMQC
6760 999644798594 HZXYQOKKYVSFYWRDGVQESZSLBDSBAMZVQBHGCBJVLHXBEUTQUTLU ,GDWSLQHIAYOYSCGDYUWZQSICFZCKGIDOSRZKEMHEIGXXTALMFFYK
3587 999637438235 FEAZIBNEGHNHUNTHQJMOZGQQSNVJJIWYACOHCCIWLEIENICHDKTHBU ,BIVXBPFXAEXDBZZXKWJNQXLDZIFTHLKFIZNHEIDMAEJQZCXVET
5614 999572594287 LXXKMUCCNZRDIFHVITYIBBTUFQUMHPRZWPQAZGKMAGBBWYWKIKNYP ,KXINQMPRNIE LUPZMHBDXYVBRGAKQYWBPTLKPWUUCFZPPVWVRFZJXJ
555 999566631984 QHQNNDALDOCIJCEVMBYFYSVKCSFNUPNGICYKKEWSPCGQHKNXYG ,XGCVTTZVTFLQVCWKMMAMZIWJPCWHDHFXFRXVUTBGYWMANKDIMMPDL
6643 999527766159 MCDUWUUHVGEENXNHJJVFXTMDGQXCYTIVFGCMHCOREYBTHNNCPR ,XKNWIXJOYYARJFTNKKZUCHTQDNXTYFNDOXAYUCZKJFSKHSSV00UB
2935 999463416692 OKVIBWGNFQSSAEHODDKHRYMXLLSLCFINEVRZIDGLWKKWXYTJZJIZ ,MMLAWCNMAMVKUOWVEMTHCOIRFAPFGOBOWMYZTTHVLOPEDVSAUC
End of Selectionsort testing.
Insertionsort testing time.....

====++ Insertion Sort (Descending) +++====
9614 999935638771 PKVYEACGIRDDIHIVSNQSTUDIINGBYZMVRFYAQWAXAYLZZDIGNMJN ,ZFBVYRDMWIEKAKSCVTSUPJSTQLAYSENVPAVHHTSXHF COHPHPQFD
3351 999923652622 YNJEGMKMIXLQMTAXIFNHDZZBNFEHSJFRJPIINSTYQSVWCEFNZGYGL ,LBXLVPMDVNLSCKKMYYDAOSADKXZBAPTZZXCLEPVVRNAFEZJNDJWH
5837 999917776745 EZJEBKRJECUONVUNNHTUQOMRKUNITKCECFBJSVXXJFDXOV CSTIBW ,WREWLUSBZZLCFWAKJBHORDHDGKQAIWDRWEXVZADKBVXDIVFKNVPC
9040 999751427723 IPXGNLCSWOSUBIOZCHVQKORTLZZYCYLQXEVZYGVBFXLUISFSPDBW ,JUKSZWKCDZDPJNNRBDHCTQKUOMVKGWNLUPQZSLHNCB EYOWVQMQC
6760 999644798594 HZXYQOKKYVSFYWRDGVQESZSLBDSBAMZVQBHGCBJVLHXBEUTQUTLU ,GDWSLQHIAYOYSCGDYUWZQSICFZCKGIDOSRZKEMHEIGXXTALMFFYK
3587 999637438235 FEAZIBNEGHNHUNTHQJMOZGQQSNVJJIWYACOHCCIWLEIENICHDKTHBU ,BIVXBPFXAEXDBZZXKWJNQXLDZIFTHLKFIZNHEIDMAEJQZCXVET
5614 999572594287 LXXKMUCCNZRDIFHVITYIBBTUFQUMHPRZWPQAZGKMAGBBWYWKIKNYP ,KXINQMPRNIE LUPZMHBDXYVBRGAKQYWBPTLKPWUUCFZPPVWVRFZJXJ
555 999566631984 QHQNNDALDOCIJCEVMBYFYSVKCSFNUPNGICYKKEWSPCGQHKNXYG ,XGCVTTZVTFLQVCWKMMAMZIWJPCWHDHFXFRXVUTBGYWMANKDIMMPDL
6643 999527766159 MCDUWUUHVGEENXNHJJVFXTMDGQXCYTIVFGCMHCOREYBTHNNCPR ,XKNWIXJOYYARJFTNKKZUCHTQDNXTYFNDOXAYUCZKJFSKHSSV00UB
2935 999463416692 OKVIBWGNFQSSAEHODDKHRYMXLLSLCFINEVRZIDGLWKKWXYTJZJIZ ,MMLAWCNMAMVKUOWVEMTHCOIRFAPFGOBOWMYZTTHVLOPEDVSAUC
End of Insertionsort testing.
Bubblesort testing time.....

====++ Bubble Sort (Descending) +++====
9614 999935638771 PKVYEACGIRDDIHIVSNQSTUDIINGBYZMVRFYAQWAXAYLZZDIGNMJN ,ZFBVYRDMWIEKAKSCVTSUPJSTQLAYSENVPAVHHTSXHF COHPHPQFD
3351 999923652622 YNJEGMKMIXLQMTAXIFNHDZZBNFEHSJFRJPIINSTYQSVWCEFNZGYGL ,LBXLVPMDVNLSCKKMYYDAOSADKXZBAPTZZXCLEPVVRNAFEZJNDJWH
5837 999917776745 EZJEBKRJECUONVUNNHTUQOMRKUNITKCECFBJSVXXJFDXOV CSTIBW ,WREWLUSBZZLCFWAKJBHORDHDGKQAIWDRWEXVZADKBVXDIVFKNVPC
9040 999751427723 IPXGNLCSWOSUBIOZCHVQKORTLZZYCYLQXEVZYGVBFXLUISFSPDBW ,JUKSZWKCDZDPJNNRBDHCTQKUOMVKGWNLUPQZSLHNCB EYOWVQMQC
6760 999644798594 HZXYQOKKYVSFYWRDGVQESZSLBDSBAMZVQBHGCBJVLHXBEUTQUTLU ,GDWSLQHIAYOYSCGDYUWZQSICFZCKGIDOSRZKEMHEIGXXTALMFFYK
3587 999637438235 FEAZIBNEGHNHUNTHQJMOZGQQSNVJJIWYACOHCCIWLEIENICHDKTHBU ,BIVXBPFXAEXDBZZXKWJNQXLDZIFTHLKFIZNHEIDMAEJQZCXVET
5614 999572594287 LXXKMUCCNZRDIFHVITYIBBTUFQUMHPRZWPQAZGKMAGBBWYWKIKNYP ,KXINQMPRNIE LUPZMHBDXYVBRGAKQYWBPTLKPWUUCFZPPVWVRFZJXJ
555 999566631984 QHQNNDALDOCIJCEVMBYFYSVKCSFNUPNGICYKKEWSPCGQHKNXYG ,XGCVTTZVTFLQVCWKMMAMZIWJPCWHDHFXFRXVUTBGYWMANKDIMMPDL
6643 999527766159 MCDUWUUHVGEENXNHJJVFXTMDGQXCYTIVFGCMHCOREYBTHNNCPR ,XKNWIXJOYYARJFTNKKZUCHTQDNXTYFNDOXAYUCZKJFSKHSSV00UB
2935 999463416692 OKVIBWGNFQSSAEHODDKHRYMXLLSLCFINEVRZIDGLWKKWXYTJZJIZ ,MMLAWCNMAMVKUOWVEMTHCOIRFAPFGOBOWMYZTTHVLOPEDVSAUC
End of Bubblesort testing.
```


Test case : ผลลัพธ์ของเวลาที่ใช้ในการเรียงข้อมูลที่ได้

```
System Can Read File = 100000
Scansort testing time....
End of Scansort testing.
Selectionsort testing time....
End of Selectionsort testing.
Insertionsort testing time....
End of Insertionsort testing.
Bubblesort testing time....
End of Bubblesort testing.

+-----+
|      Sort      | Random data (n) | Insert data(n+1) | Descending data(n+1) |
+-----+
| Scan sort      | 196.432000 ms | 111.715900 ms | 254.868601 ms |
| Selection sort  | 74.106400 ms | 67.992700 ms | 155.185801 ms |
| Insertion sort  | 77.282600 ms | 0.169700 ms | 150.393400 ms |
| Bubble sort    | 205.693300 ms | 0.259500 ms | 220.968400 ms |
+-----+

Program is written by สรร แก้วโชติช่วงกุล 63070501067 CPEREGULAR
```

=== สรุปผล ===

Scan sort : เวลาที่ใช้จากการเรียงลำดับแบบ Random, Insert และ Descending มีค่าประมาณ 196,112 และ 255 ms. ตามลำดับ

Selection sort : เวลาที่ใช้จากการเรียงลำดับแบบ Random, Insert และ Descending มีค่าประมาณ 74,68 และ 155 ms. ตามลำดับ

Insertion sort : เวลาที่ใช้จากการเรียงลำดับแบบ Random, Insert และ Descending มีค่าประมาณ 77,0.17 และ 150 ms. ตามลำดับ

Bubble sort : เวลาที่ใช้จากการเรียงลำดับแบบ Random, Insert และ Descending มีค่าประมาณ 206,0.26 และ 221 ms. ตามลำดับ

เวลาที่ใช้ ในการเรียงลำดับแบบต่างๆจะสรุปได้ดังนี้

Random data(จากน้อยไปมาก) : Selection sort < Insertion sort < Scan sort < Bubble sort

Insert data(จากน้อยไปมาก) : Insertion sort < Bubble sort < Selection sort < Scan sort

Descending data(จากมากไปน้อย) : Insertion sort < Selection sort < Bubble sort < Scan sort

สรุป : การเรียงจากน้อยไปมากแบบ Random data ➡ Selection sort เร็วที่สุด

การเรียงจากน้อยไปมากแบบ Insert data ➡ Insertion sort เร็วที่สุด

การเรียงจากมากไปน้อยแบบย้อนกลับ Descending ➡ Insertion sort เร็วที่สุด

- สรุปความเข้าใจของตนเอง

ใน Assignment นี้เราต้องนำข้อมูลที่ได้จากการอ่านจากไฟล์มาทดสอบเรียงลำดับ 3 แบบ Random Insert และ Descending และหาเวลาในการเรียงลำดับแบบต่างๆโดยมีวิธีการเรียงลำดับอยู่ 4 วิธีคือ Scan sort, Selection sort, Insertion sort และ Bubble sort และนำมาแสดงผลดูว่าการเรียงลำดับแบบไหนใช้เวลาน้อยที่สุดและสรุปผล

- ผลการประเมินตนเอง

ให้ตนเองอยู่ที่ระดับ 80 เพราะสามารถทำโจทย์ได้ด้วยตัวเองและมีปัญหาเล็กน้อยในส่วนของการสร้าง method ของแต่ละ sort และยังไม่มั่นใจ 100 % ในการทำโจทย์ประยุกต์จากโจทย์นี้

160

63070501067
SORATHORN
KAEWCHOTCHUANGKUL

Grading Rubric

View Full Rubric

Criterion 1

20

40

60

80

100

80

การเขียนโปรแกรมด้วย Algorithms ที่กำหนดให้

20

40

60

80

100

80