"EEM 480 Algorithms and Complexity" Homework #1 File Reading and Sorting

Mehmet Çağrı Aksoy

In this homework, I investigated how to take data from text, how to save them to array, type conversion functions problems, shorting algorithms and complexity. During my trials I have had many errors and problems, I solved them thanks to stackoverflow and EEM480 course slides.

I have used Netbeans IDE 8.2 for working environment.

First, I'd like to introduce my program. My program wants to file path from keyboard. After you have wrote the path, try catch algorithm is running, if path is true, everything is OK, so we can pass to next step, otherwise, "catch" part of algorithm gives error message on the screen and call itself again (recursive).

Second step of program save data from text to arrays. As we know our database has different types of variables. For example, IDs has long but Name and surname has String type. When I get the data from text, it accepts all of them as string. I have used;

```
int result = Integer.parseInt(MyString);
```

to convert string to Integer, also we have functions that converts string to long which is (Long.ParseLong).

As my database provide, first line of database gives us number of people which is in database. I used;

```
arrsize = readCodes.nextInt();
```

It reads integer value from text file. It will usable when we want to sort people.

The last but the most important part is sorting part. As I learned in the lecture, I have many options which are "Bubble, Insertion, selection sorting etc.". I have used Bubble sort, I don't know but It sounds nice 😊 Bubble sort 😊

```
for (int i = stu.arrsize; i > 1; i--){
            for(int j = 0; j <i-1; j++){
                if(stu.idarr[j]>stu.idarr[j+1]){
                    long dummy = stu.idarr[j];
                    stu.idarr[j]=stu.idarr[j+1];
                    stu.idarr[j+1]=dummy;
```

Bubble sorting algorithm can be seen here. I used this algorithm in my code block a lot.

Let's talk about a little bit Complexity. When I investigated some Big-O complexities of common algorithms used in computer science and jave I have found some functions. As I mentioned before I have used bubble sort whose complexity is `Ω(n) or Θ(n^2)`. Also, during my work, I worked Arrays, array copy, get data from array etc. Let's see array complexities;

Common Data Structure Operations

Data Structure	Time Complexity							
	Average				Worst			
	Access	Search	Insertion	Deletion	Access	Search	Insertion	Deletion
Array	Θ(1)	Θ(n)	Θ(n)	Θ(n)	O(1)	O(n)	O(n)	O(n)
Stack	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)
Queue	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)
Singly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)
Doubly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)

Figure 1-Common Data Structure's Complexities

Although the bubble sort algorithm is not the best sorting method in terms of complexity, speed and time usage as seen in the table, I used it in my homework. Because it's easy to use.

Array Sorting Algorithms

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
Quicksort	Ω(n log(n))	Θ(n log(n))	O(n^2)	O(log(n))
Mergesort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(n)
Timsort	Ω(n)	Θ(n log(n))	O(n log(n))	O(n)
Heapsort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(1)
Bubble Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Insertion Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Selection Sort	Ω(n^2)	Θ(n^2)	O(n^2)	O(1)
Tree Sort	Ω(n log(n))	Θ(n log(n))	O(n^2)	O(n)
Shell Sort	Ω(n log(n))	Θ(n(log(n))^2)	O(n(log(n))^2)	O(1)
Bucket Sort	Ω(n+k)	Θ(n+k)	O(n^2)	O(n)
Radix Sort	Ω(nk)	Θ(nk)	O(nk)	O(n+k)
Counting Sort	Ω(n+k)	Θ(n+k)	O(n+k)	O(k)
Cubesort	Ω(n)	Θ(n log(n))	O(n log(n))	O(n)

Figure 2-Sorting Algorithm'S Complexities and Big-O notations