HACETTEPE UNIVERSITY DEPARTMENT OF COMPUTER ENGINEERING
BBM204

Prepared by OKAN ALAN Number : 21526638
E-Mail : okanalan97.14@gmail.com

Subject : Assignment 1

Problem

We have some records about network communication. These recorded by time. These have more than 80 features.

What will we do these data? We will sort these records by given column number. Meanwhile we should calculate which algorithm how much time used. After that we analyze results of time and memory complexities.

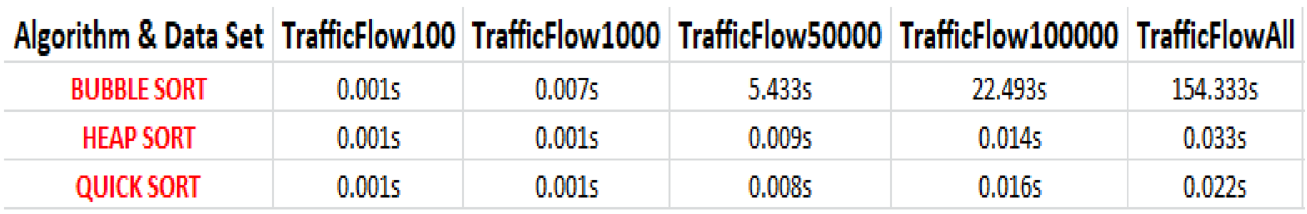
I was learned by the assignment when or where to use which algorithm. Reading and writing to any file is slower than sorting. It is funny but real.

Solution

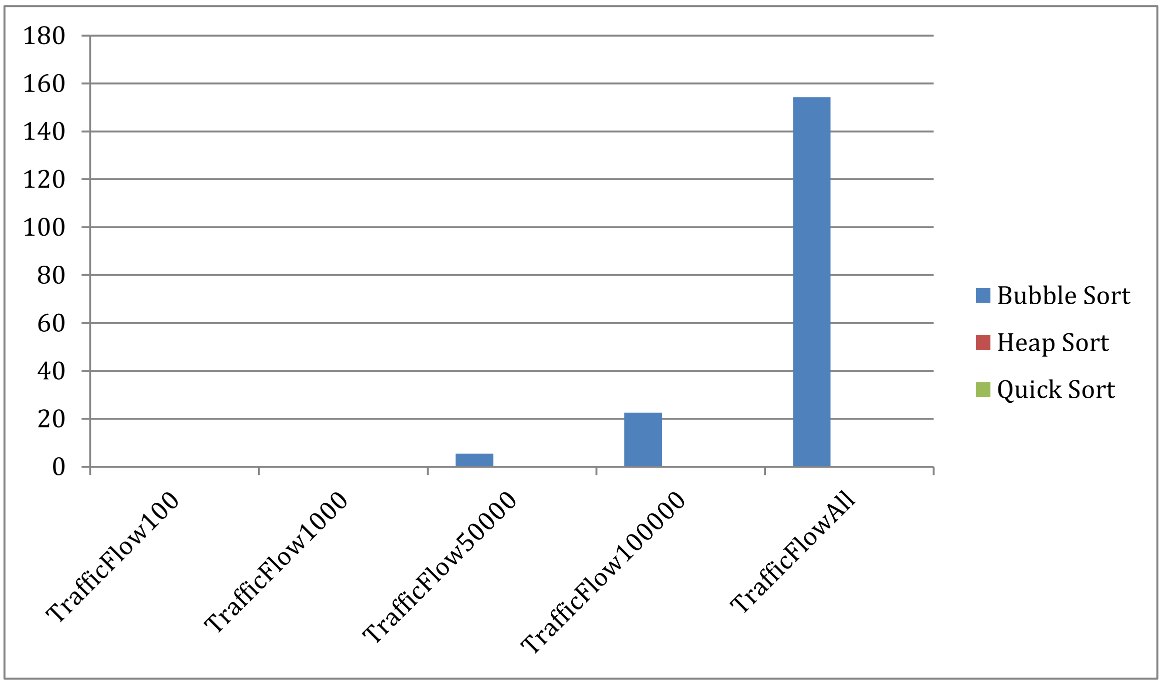
The assignment want us to choose 3 different sorting algorithms. I chose bubble, quick and heap sort algorithms.

Expecting From Us

Expecting is analysis of sorting algorithms. Analysis of algorithms depends on these parameters; how much time, data, space and memory we have. Finding which algorithm the most efficient at which situation. This assignment give a big lesson. If you are bad programmer and you write O(n^2) algorithm, you will wait until die.



Algorithm & Data Set	TrafficFlow100	TrafficFlow1000	TrafficFlow50000	TrafficFlow100000	TrafficFlowAll
BUBBLE SORT	0.001s	0.007s	5.433s	22.493s	154.333s
HEAP SORT	0.001s	0.001s	0.009s	0.014s	0.033s
QUICK SORT	0.001s	0.001s	0.008s	0.016s	0.022s



Algorithm

First of all I started thinking about how do I store the records. I decide I will store these at ArrayList after reading from csv file. All data kept at “table” variable as a String. Then I sorted records according to column number. I tried 2 type of sort. First I sent all of data to sort and It is slower than other because I need more space, memory and time when I swap line during sorting. That’s why I implement second. It is different from first due to sending items. This time I sent the data at the given column with order to function.

Sending list could be create 2 different way. First is that I can control which column now while I reading csv file and I can added data to sending list which is from given column number. Every time controlling column it is not good. It is complexity “84*countLines”. Second time I store all of them after that I get the given column. This is only “countLines”. Second is faster than first. Then I sent list to function to sort data. Sent list name is “going(Quick,Heap,Buble)”.

Finally I had sorted data. I looked saving option to save or not. If it is T I would overwrite to read csv file. Writing algorithm is a bit tricky. I get element at sorted sent list in turn. Every element contains “sequence and data”. I take line from “table” with learnt “sequence”. Then I wrote to file.

Sorting Algorithms

*Bubble

Bubble sort’s time complexity is average O(n^2). If you don’t have hundreds of data it will be good. It is slow because I take one element from the list in order and compare it with all the rest. Isswaped variable control list. If last comparing loop list is not swapped, this variable will break bubble sort. This variable saves our time.

*Heap

Heap sort’s time complexity is average O(n*logn). It is look like selection sort but it is clever than selection sort. The reason for being clever is that selection find min/max element O(n) heap O(1). In this assignment Heap Sort consist 2 different function. They are *sort* and *heapify*. *Heapify* control a array to be max heap array.If you say why max heap,I will need sorted list. I send biggest element to end of the array during sorting list and then I get sorted array.
Sort is created by 2 for loop. First loop look first (n/2)-1 indices to check max heap. First (n/2)-1 indices have child. N is length of the array. Second loop is part of deleting.

*Quick

Quick sort’s time complexity is average O(n*logn). This is different from others. This has “Pivot player”. This compare element to Pivot and partition list. In my homework partition is at end of quicksort function because of stackover flow error. This reason is a lot of recursion of 2 different function.

*Memory

If I am not wrong, memory usage happens only during swaping and recursion.