John Hopkins University

605.202 Data Structures

Obadiah K Njoroge

Lab 4 Analysis

12/07/2021

**Introduction**

Sorting is the process of ordering or re-ordering data elements in a predefined format or sequence, for example ascending or descending sequences. Sorting data improves the efficiency of processes performed on that data such as insertion, deletion, searching and storage.

Sorting algorithms define how the data elements are arranged. Some examples of sorting algorithms include quick sort, and heap sort. This analysis will discuss these two sorting algorithms and look at their asymptotic notation/cost which is a measure of the speed and efficiency of an algorithm as the size of the dataset increases.

Quicksort is a highly efficient in place sorting algorithm that can also be descried as a divide and conquer algorithm. It works by partitioning a list / array into sub lists using a specified value, otherwise known as a pivot. It then calls itself recursively to sort the sub arrays. The pivot is compared to the elements within the file and move the smaller elements in the left of the pivot and larger elements on the right side of the pivot. This process is repeated until the array is fully sorted. On best case, the time complexity for a quicksort algorithm is O(n log n) while the worst-case scenario would be O(n2).

Heap sort is a type of comparison sorting algorithm that is based on binary heap where each node is greater (max heap) or smaller (min heap) than its descendants. Heap sort is performed in a complete tree. This allows for the smallest possible height, and therefore favors time complexity.

When using heap sort, the first step is to build a max or min heap then heapifying the elements to meet the heap properties.

**Quick sort vs Heap sort**

Some of the similarities between quicksort and heapsort include:

1. They are both fast algorithms. The best and average cases for quick sort are O(n log n). Heapsort has a similar time complexity running at O(n log n). Quicksort however can take O(n) in a worst-case scenario.

Differences

1. The main difference between quicksort and heapsort is that quicksort runs a recursive operation of the sorting list while a heapsort is based on heap data.
2. Heapsort is more memory efficient than quick sort algorithm.

**Design Justification**

The implementation of this lab required writing two separate sorting algorithms i.e., quicksort and heapsort and comparing their asymptotic cost. To compare the two algorithms, 15 files were created of varying sizes. The files were divided into three groups: randomly orders, reverse ordered and finally ascending order. Each of three groups contains 5 files hold size inputs: 50, 1000, 2000, 5000 and 10000 integers.

Each of the files was ran through each of the algorithm and the output stored in a output folder within a file that shared the same name as the input file. Therefore input/asc\_1K.dat was stored in output/acs\_1K.dat.

**Challenges encountered**

The main challenge encountered was efficiently collecting data to analyze the number of steps each algorithm took to process each of the three different files i.e. reverse, ascending and randomly ordered files.

**What I might do different next time**

If I was to re-do this project again, I would find a better way to analyze the asymptotic cost and potentially graph it for a better visual aid. I would also create two output folders for each of the sorting algorithms for a better illustration of the output of each algorithm. Currently the outputs are held on one file and depending on the file size, it might be difficult to read the outputs.

**Improvements/Enhancements**

I added one more file with 20000 integers to further compare the asymptotic cost.

References

Rajinikanth. “Data Structures.” *Data Structures Tutorials - Heap Sort Algorithm*, http://www.btechsmartclass.com/data\_structures/heap-sort.html.