DOKUZ EYLUL UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

CME 2204 ALGORITHM ANALYSIS

ASSIGNMENT-1 REPORT

COMPARISON OF HEAP SORT, DUAL PIVOT QUICK SORT AND SHELL SORT

By

Nilay YÜCEL - 2017510082

April, 2020

İZMİR

CONTENTS

CHAPTER ONE

INTRODUCTION

This report is the report of Assignment-1 given in the CME 2204 Algorithm Analysis course. Assignment-1 was given on March 13th and is a 5-week assignment. Its purpose is to compare *heap sort*, *dual pivot quick sort* and *shell sort* algorithms. We are expected to run and analyze these algorithms for different scenarios.

"Heap Sort" is a comparison based sorting technique based on "Binary Heap" data structure. The largest item is stored at the root of the heap. "Dual Pivot Quick Sort" is to take two pivots, one in the left end of the array and the second, in the right end of the array. Then, it partitions the array into three parts. In the first part, all elements will be less than the left pivot, in the second part all elements will be greater or equal to the left pivot and also will be less than or equal to the right pivot, and in the third part all elements will be greater than the right pivot. "Shell Sort" is mainly a variation of "Insertion Sort". In "Shell Sort", elements at a specific interval are sorted. The interval between the elements is gradually decreased based on the sequence used.

“Heap Sort”s best case and worst case are the same and it is O (nlog (n)). The best case of “Dual Pivot Quick Sort” and “Shell Sort” are O (nlog (n)) and worst case is O (n ^ 2). These worst cases and best cases behave according to the given array.

In accordance with this assignment, the *table-1* given on page 3 was created. Necessary explanations are made in chapter two.

CHAPTER TWO

EXPLANATION

	EQUAL INTEGERS			*RANDOM INTEGERS*			*INCREASING INTEGERS*			*DECREASING INTEGERS*		
	1,000	10,000	100,000	1,000	10,000	100,000	1,000	10,000	100,000	1,000	10,000	100,000
heapSort	0,2667	1,4576	9,7116	0,7571	4,6673	24,6928	0,7553	4,8806	20,7218	0,7715	4,3372	18,7405
dualPivot QuickSort	0,8353	2,5006	14,0057	0,6115	3,0541	22,2581	5,8248	23,8814	1330,173	5,1895	24,4213	1311,3924
shellSort	0,3033	3,0669	12,6924	0,8965	6,5027	26,5415	0,2582	5,9613	12,3904	0,6902	5,3818	15,3961

Table 1- *It contains the runtime of algorithms in ms. Turquoise color shows the time of the fastest running algorithm for the given array.*

In this chapter, algorithms are explained over table-1 prepared according to Assignment-1.

When the size of the given array grows, the run time of the algorithms increases. It can observe from Table-1. Algorithms given for different arrays give different time results. "Heap Sort"s best case is equal integers. "Dual Pivot Quick Sort"s best case is random integers in arrays with 1,000 size, in other sizes is equal integers. its worst case is increasing and decreasing intergers, "Shell Sort"s best case is increasing intergers in arrays with 1,000 and 100,000 size, in 10,000 size is equal integers. Its worst case appears to be in random intergers.

If we look at from a different perspective, the best algorithm to use for equals intergers is “Heap Sort”. It is best to use "Dual Pivot Quick Sort" for random integers. For increasing and decrasing integers, "Shell Sort" should be used for arrays with 1000 and 100000 size. However, "Heap Sort" should be used for arrays with 10000 size.

CHAPTER THREE

SCENARIO

“ We aim to place students at universities according to their central exam grades and preferences. If there are millions of students in the exam, which sorting algorithm would you use to do this placement task faster?”

In the first stage, we know that the grades are random when sorting by grades. We mentioned in chapter 2 that "Dual Pivot Quick Sort" is the best algorithm in random numbers. It is the most effective method to use "Dual Pivot Quick Sort" in the random intergers arrays that are large size, because it sorts both from the beginning and from the end.

In the second stage, the algorithm that should be used in preferences is "Shell Sort". Numbers are used sequentially in preferences and "Shell Sort" gives the best result in increasing or decreasing intergers in big size."Shell Sort" is the best for sequent integers because in "Shell Sort", elements at a specific interval are sorted.

CHAPTER FOUR

OVERALL ASSESSMENT OF THE ASSIGNMENT-1

In this report, Assignment-1 given in chapter one is explained. Also necessary information about "Heap Sort", "Dual Pivot Quick Sort" and "Shell Sort" are given. Chapter two run time analysis was done and table-1 was interpreted. The scenario given in Chapter three was interpreted and the best solution was given.

As a result of the Assigment-1 given, it was learned that "Heap Sort", "Dual Pivot Sort" and "Shell Sort" algorithms and which algorithm would be better for different scenarios.

CHAPTER FIVE

REFERENCES

- http://bilgisayarkavramlari.sadievrenseker.com/2008/12/20/kabuk-siralama-shell-sort/
- http://bilgisayarkavramlari.sadievrenseker.com/2008/08/09/yiginlama-siralamasi-heap-sort/
- https://www.programiz.com/dsa/shell-sort
- https://www.programiz.com/dsa/heap-sort
- https://en.wikipedia.org/wiki/Shellsort
- https://www.geeksforgeeks.org/heap-sort/
- https://www.geeksforgeeks.org/dual-pivot-quicksort/
- https://www.geeksforgeeks.org/shellsort/