Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

3D5A DATA STRUCUTRES AND ALGORITHM

Assignment 2 Report

Prepared By:

PULKIT SHARMA

19323659

Computer Engineering

Task 1

Implement quicksort for sorting an array of integers. Your implementation should count the number of times values are swapped in the array and the number of times two values in the array are compared.

- Quicksort is a divide and conquer algorithm which divides the arrays into smaller portions and sorts each part. Therefore, the swaps and comparisons are lesser than the other algorithms. The value of pivot element changes the result.
- We get to observe that during the best case i.e. ascending order there are **0** swaps with **50004999** comparisons whereas the worst case i.e. descending it is **5000** swaps which was expected as the number of swaps in quicksort for descending is exactly the half.
- In random values we observe that the swaps are still less considering the large size of the array. **29651** swaps are still better than the algorithms that do not use divide and conquer algorithm.
- In uniform list of values, we observe that there are **0** swaps that is rightfully so because of the values being same.

Task 2

Implement a sorting algorithm of your choice and evaluate it in the same manner as you did quicksort (i.e. print swaps/comparisons and test for each of the 5 different types of data). Does the algorithm you picked perform better or worse under each of the five conditions?

```
TEST  : Unique random values
SORTED: Y
SWAPS : 24924873
COMPS : 24934860
--------------------------------------------------

TEST  : Random values
SORTED: Y
SWAPS : 24858458
COMPS : 24868453
--------------------------------------------------

TEST  : Ascending sorted list
SORTED: Y
SWAPS : 0
COMPS : 9999
--------------------------------------------------

TEST  : Descending sorted list
SORTED: Y
SWAPS : 49995000
COMPS : 49995000
--------------------------------------------------

TEST  : Uniform list
SORTED: Y
SWAPS : 0
COMPS : 9999
--------------------------------------------------
```

- Insertion sort is a simple sorting algorithm that builds the final sorted array one item at a time. It iterates the input elements by growing the sorted array at each iteration.
- We get to observe that during the best case i.e. ascending order there are **0** swaps with **9999** comparisons whereas the worst case i.e. descending it is **49995000** swaps and comparisons which is expected as all the values are to be changed in the array.
- In random values we observe that the swaps are very large because of the large size of the array. **24868453** swaps and roughly same number of swaps is a lot higher than the quicksort algorithm.
- In the uniform list we get the expected values.
- This algorithm is inefficient when performed on large size of array. There is only one case where we get less values and that is ascending because it just goes through the array once. This algorithm is not used in real life applications due to these limitations. We conclude that quicksort algorithm is better than insertion sort algorithm.

Task 3

You have been provided with a dataset of game reviews which have been gathered from IGN over the last 20 years. Sort the reviews on the basis of game scores and find out what the most popular games of the last 20 years are.

Title	Platform	Score	Release Year
Diablo III	PC	10	2012
Rayman 2: The Great Escape	Dreamcast	10	2000
The Lord of the Rings: The Return of the King	Wireless	10	2003
Devil May Cry 3: Dante's Awakening	PlayStation 2	10	2005
Devil May Cry	PlayStation 2	10	2001
Device 6	iPhone	10	2013
Ratchet & Clank: Up Your Arsenal	PlayStation 2	10	2004
Rayman Legends	Xbox 360	10	2013
Rayman Origins	Wii	10	2011
Digital Chocolate Cafe	Wireless	10	2007

- We first loaded the ign.csv filed and stored it in structure. Then we sorted it based on score and displayed the Top 10 videogame of last 20 year with 10 score.

2. To Display top 5 game of each year what we can do is sort the game on basis of year first and then a different function which would further sort the stored values based on the score. Then we can show the top score games of each year with the addition of this new function.