Sort ALgorithms Analysis

This is a brief documentation of of the Visual Studio project and the accompanying Python script that uses Matplotlib library to create plot charts of the sorting runtime.

Visual Studio Project

The Visual Studio Project consists of different folders, the most important is *Algos* containing 3 main file: - Algos.h - Algos.cpp - Main.cpp

With all member functions declaration in *Algos.h* and definition in *Algos.cpp*. *Main.cpp* is where Algos class function are used to obtain data sets from two file: - input_large.txt - input_small.txt

and sorting runtimes saved in: - sortResults.txt - sortResults_milliseconds.txt

Template classes are used to enable dealing with data sets of different data types.

Algos.cpp

generateArray()

- Generates and returns an vector of random elements depending on the size and maximum element value given as parameters.
- Time Complexity: O(N)

recordArray()

- Takes a vector generated by `generateVector()` and writes it to one of two `.txt` files, depending on the vector size. The two files are `input_small.txt` and `input_large.txt`.
- Time Complexity: O(N)

recordTime()

- Takes a string with the sort name and a vector of pair with the data set sizes and the time taken by the sorting algorithm for each one, then saves the time to `sortResults.txt` file.
- Time complexity: O(N)

fetchArray()

- Takes an int as a parameter determining of the size of the array to fetch from the data files and setting the private member `myArr` equal to that array.
- Time Complexity: O(N), where N is the file size.

MergeSort()

- Takes an array as a parameter and sorts the array using *Divide and Conquer* method. The array is split into two halves recursively and then each two halves get merged using `Merge()`.
- Time Complexity: O(NLog(N))

Merge()

- Helper function for `MergeSort()`.
- Takes 3 integer parameters determining the two halves to be merged.
- The function uses an auxillarry array to put each element of the two halves into place.
- Time Complexity: O(N)

QuickSort()

- Takes an array as a parameter and sorts the array using Divide and Conquer method.
- The array is split into two halves using `partition` where each half should contain all elements smaller than a choosen pivot and other half containing all elements bigger than the pivot.
- Time Complexity: O(NLog(N)) (On Average).

partition()

- Takes two integer parameters determining the begin and end of the sub-array to be partitioned.
- The pivot element is choosen as the last element in the sub-array
- Returns the index at which the array is partitioned.
- Time Complexity:O(N).

AlgosTimePlots.py

`AlgosTimePlots.py` uses *Matplotlib* library to create plots for each sorting algorithm runtime against several data sets.

- The Python script uses `sortsResults_milliseconds.txt` previously populated file to create a plot of each sorting algorithm performance against time.
- Charts are saved in the same `Algos Analysis` directory.