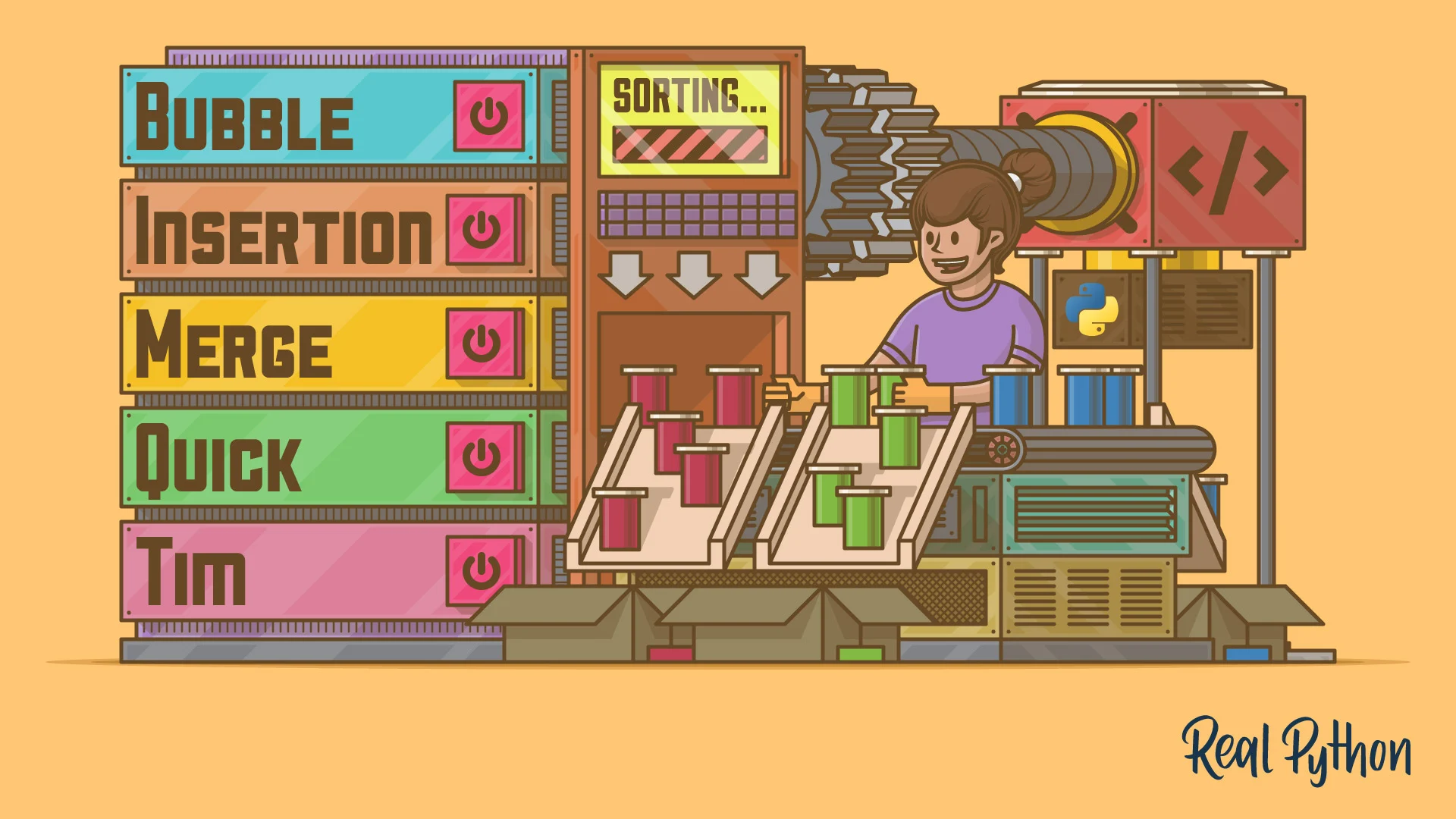
OS PROJECT REPORT

*Submitted to Ms Tania Iram & Ms. Anaum Hamid*



**Group Members:  
BCS-4D**

**Noor Fatima (20K-0406)  
Arooba Moin (20K-0213)  
Bismah Akram(20K-0449)**

Department of Computer Science

National University of Computer and Emerging Sciences-FAST

Karachi Campus

**Parallel Programming Comparison of Sorting Algorithms using Pthreads vs OpenMP vs Serial [3 Algorithms]**

INTRODUCTION

This project is part of our Operating Systems Course, we chose to work on the topic of 'Parallel Programming Comparison of Sorting Algorithms Using Pthreads vs OpenMP vs. Serial' using heap sort, radix sort and count sort.

We implemented a system that allowed us to differentiate the time required to execute different sorting algorithms using several approaches such as Pthreads, OpenMP for Multithreading, and Serial processing. The execution time was then analyzed of each algorithm using bar graphs which we plotted using pyplot on Google Colab to determine which is the most efficient.

METHODOLOGY

In this project, we have used the execution models; Serial, OpenMP, & PThreads where serial refers to executing one computation at a time and parallelism refers to the simultaneous execution of several calculations. Pthreads is an execution model that allows a program to control multiple different flows of work that overlap in time. OpenMP is an application programming interface for parallelizing sequential programs written in C on shared-memory platforms. It supports loop-level and function-level parallelism.

FEATURES

The main function calls a function to generate an array randomly dynamically. Then this array is sorted using sorting algorithms of Heap, Radix and Count Sort. Each is sorted 3 times using different methods of executions of Pthreads, Serial and OpenMP. For each method, time is computed and this is then compared using a graph in Python.

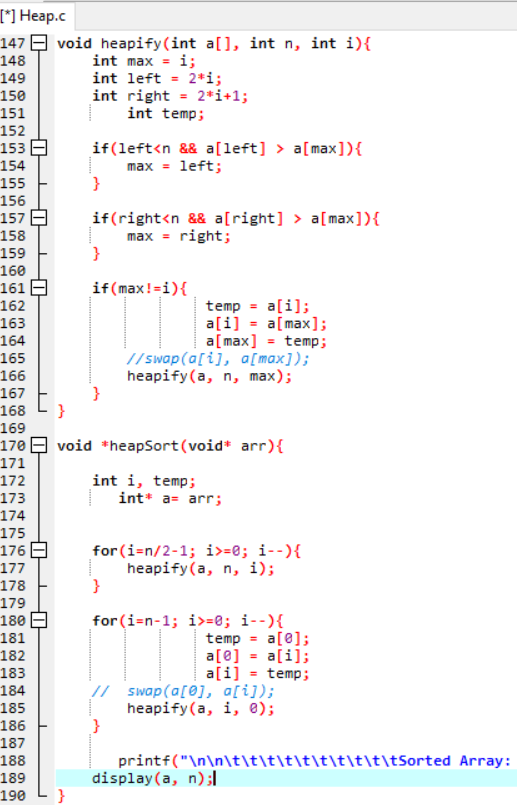
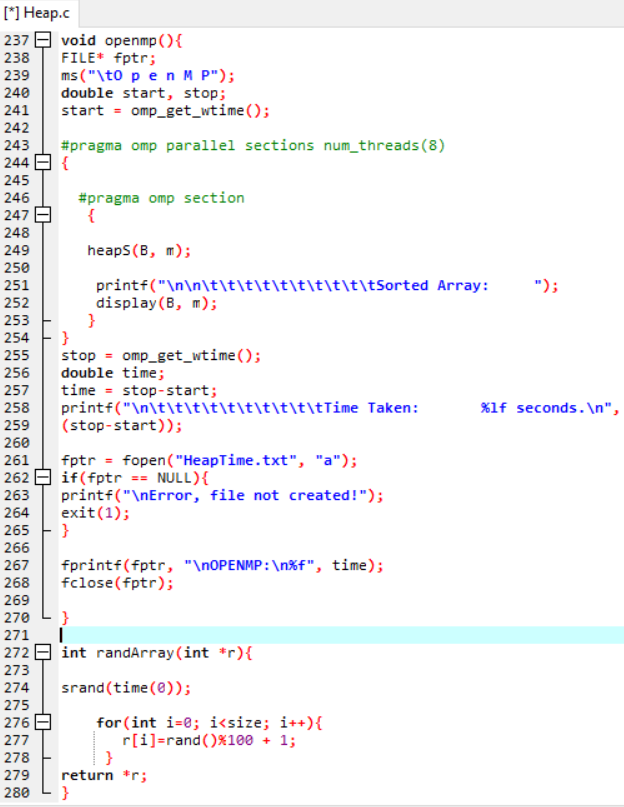
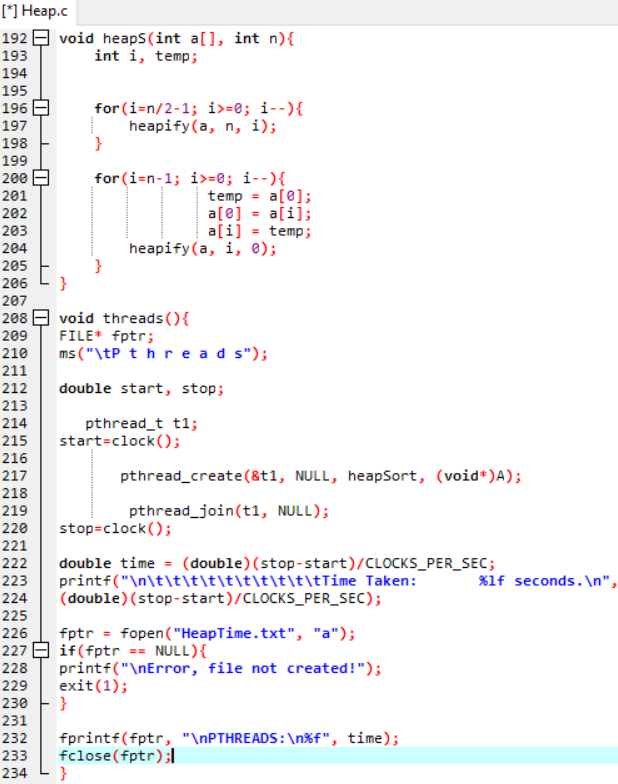
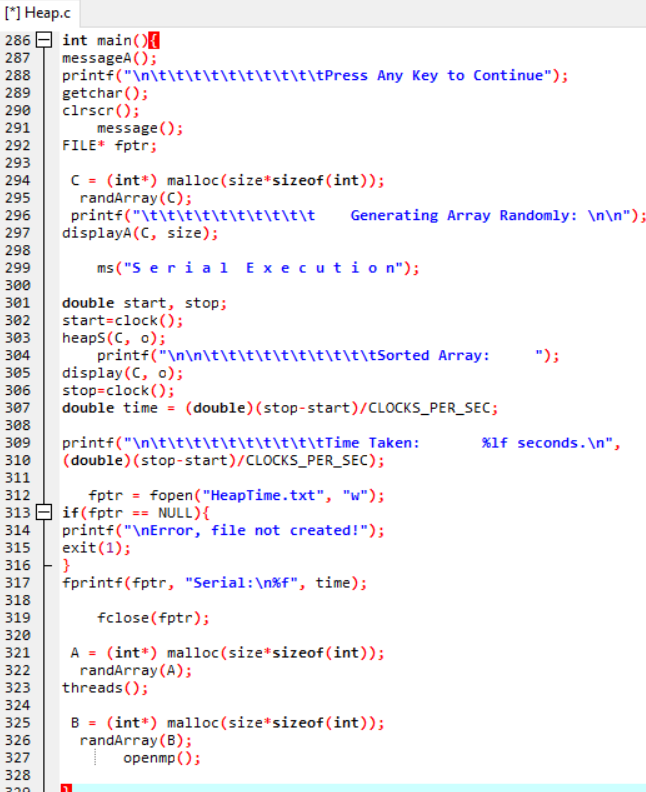
The tools we have utilized are as follows:

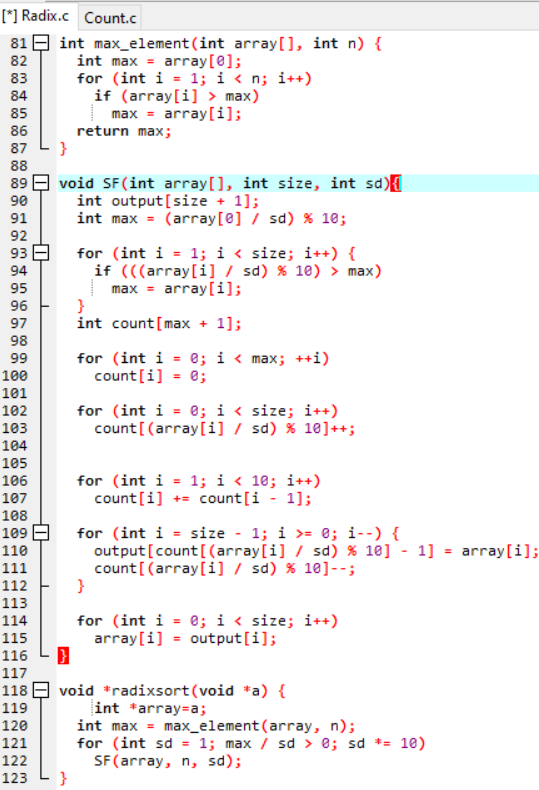
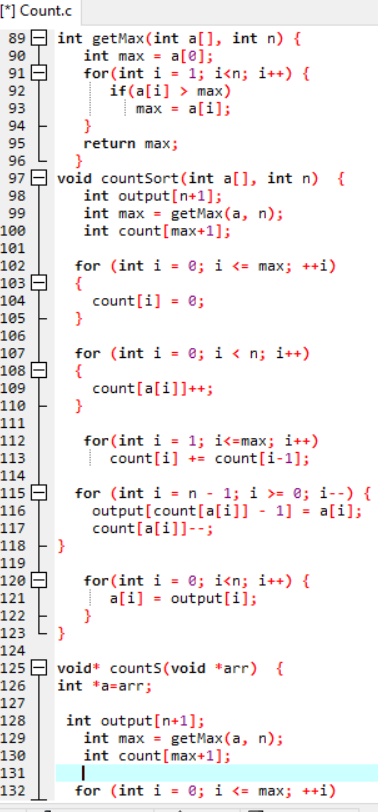
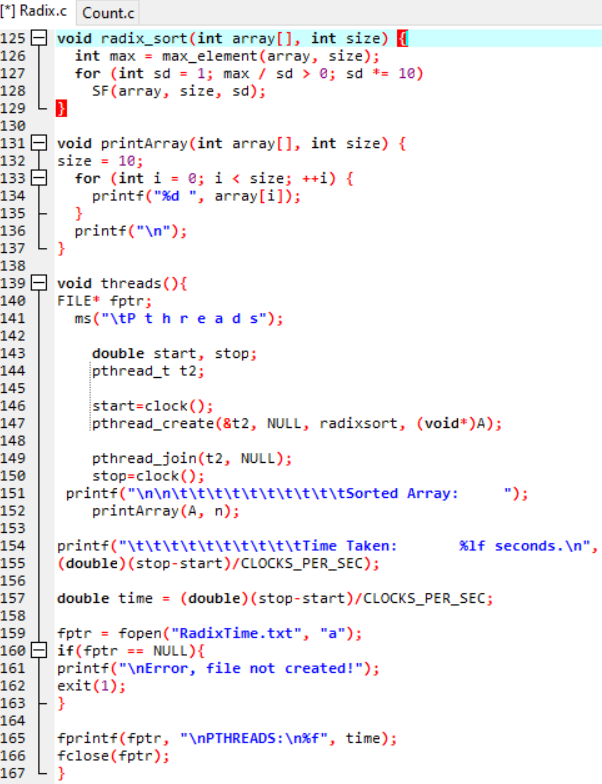
● C language

● Google Colab

● Linux (Ubuntu)

Code SNIPPETS

OUTPUT SNIPPETS



