**Bilkent University CS 342 Project 1 Report**



BILKENT UNİVERSİTY

EGE TURAN – 21502441

CS 342 – SEC1

FALL-2019

Table of Content

[1. Runtime Results for the Experiment. 3](#_Toc23365102)

[1.1 Part1 3](#_Toc23365103)

[1.2 Part2 3](#_Toc23365104)

[1.3 Part3 4](#_Toc23365105)

[2. Explanation for the data. 5](#_Toc23365106)

# Runtime Results for the Experiment.

## 1.1 Part1

File Size: 10000

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 5 | 1 | 10000 | 0.000169 |
| 5 | 2 | 10000 | 0.000246 |
| 5 | 3 | 10000 | 0.000271 |

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 500 | 1 | 10000 | 0.001021 |
| 500 | 2 | 10000 | 0.002102 |
| 500 | 3 | 10000 | 0.004413 |

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 1000 | 1 | 10000 | 0.010401 |
| 1000 | 2 | 10000 | 0.007996 |
| 1000 | 3 | 10000 | 0.014922 |

File Size: 100000

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 5 | 1 | 100000 | 0.000240 |
| 5 | 2 | 100000 | 0.000314 |
| 5 | 3 | 100000 | 0.000387 |

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 500 | 1 | 100000 | 0.001059 |
| 500 | 2 | 100000 | 0.002249 |
| 500 | 3 | 100000 | 0.003861 |

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 1000 | 1 | 100000 | 0.003327 |
| 1000 | 2 | 100000 | 0.007501 |
| 1000 | 3 | 100000 | 0.014774 |

## 1.2 Part2

File Size: 10000

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 5 | 1 | 10000 | 0.000681 |
| 5 | 2 | 10000 | 0.000558 |
| 5 | 3 | 10000 | 0.000493 |

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 500 | 1 | 10000 | 0.002580 |
| 500 | 2 | 10000 | 0.011976 |
| 500 | 3 | 10000 | 0.007443 |

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 1000 | 1 | 10000 | 0.005122 |
| 1000 | 2 | 10000 | 0.007897 |
| 1000 | 3 | 10000 | 0.055647 |

File Size: 100000

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 5 | 1 | 100000 | 0.000366 |
| 5 | 2 | 100000 | 0.000653 |
| 5 | 3 | 100000 | 0.001138 |

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 500 | 1 | 100000 | 0.002415 |
| 500 | 2 | 100000 | 0.003380 |
| 500 | 3 | 100000 | 0.014246 |

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 1000 | 1 | 100000 | 0.009242 |
| 1000 | 2 | 100000 | 0.022203 |
| 1000 | 3 | 100000 | 0.045225 |

## 1.3 Part3

File Size: 10000

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 5 | 1 | 10000 | 0.206944 |
| 5 | 2 | 10000 | 0.408023 |
| 5 | 3 | 10000 | 0.614875 |

Run-time changes for fix size K and File Size.

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 500 | 1 | 10000 | 0.203156 |
| 500 | 2 | 10000 | 0.411227 |
| 500 | 3 | 10000 | 0.622320 |

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 1000 | 1 | 10000 | 0.205939 |
| 1000 | 2 | 10000 | 0.408970 |
| 1000 | 3 | 10000 | 0.635215 |

File Size: 100000

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 5 | 1 | 100000 | 23.719320 |
| 5 | 2 | 100000 | 48.946834 |
| 5 | 3 | 100000 | 73.561023 |

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 500 | 1 | 100000 | 23.404712 |
| 500 | 2 | 100000 | 47.480775 |
| 500 | 3 | 100000 | 72.579610 |

|  |  |  |  |
| --- | --- | --- | --- |
| K | N | File sizes | Run-time |
| 1000 | 1 | 100000 | 23.852027 |
| 1000 | 2 | 100000 | 49.771508 |
| 1000 | 3 | 100000 | 74.716651 |

# Explanation for the Data.

I used Virtual Machine while testing and implementing the project. Virtual machine uses multi-processing while computing. My virtual machine uses 5 CPU to compute. This is the first reason why execution time does not increase linearly with N (number of processes) with small input sizes because child processes can be executed in different CPUs concurrently.

On the other hand, giving tremendous amount of inputs to the program -for example 1000000 input – takes lots of time because I used bubble sort algorithm to sort the inputs. Because running time of bubble sort is N2, I could not able to calculate running time with 1000000 inputs.

As I expected, changing the input K does not affect the running time of the program that much because writing K input into file does not require so much time.

I inspect that using message queue is much faster than using file storage. It makes sense to me because message queue creates shared storage area and it provides better time consumption. Another process can read that process area. As we can see in the tables, message queues are really fast because using file storage need kernel support which takes time. Without a kernel support we can read