**Program Structures & Algorithms**

**Spring 2022**

**Assignment No. 4**

Name: Hung Chih Huang

(NUID): 001005756

* **Task: Parallel Sorting**
* **[Program]  
  [Main.java]**

**Text

Description automatically generated**

**[PartSort.java]  
Text

Description automatically generated**

**Text

Description automatically generated**

**[Report]**

1. For variable array size :

* The larger cut-off, the more efficiency
* Ideal cutoff can be concluded as little more than 10 % of the array element which generates threads as well as takes less time in sorting
* The observations are in the excel file named ‘assignment4\_results.xlsx in the project directory. Another file ‘assignment4\_results.csv’ is generated by the professor program.

Graphical user interface, table

Description automatically generated

1. For variable array size :

when we try to use the parallel sort technique, the best cut off value, based on multiple runs and taking an average is - 300

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CutOff | Try1 | Try2 | Try3 | Try4 | Try5 | Try6 | Try7 | Try8 | Try9 | Try10 | Average |
| 1200 | 4 | 5 | 5 | 6 | 6 | 5 | 4 | 5 | 6 | 5 | 5.2 |
| 1000 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 1.3 |
| 500 | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 1 | 1.2 |
| 300 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0.4 |
| 200 | 6 | 8 | 13 | 4 | 3 | 3 | 5 | 2 | 11 | 3 | 5.8 |
| 100 | 15 | 13 | 12 | 12 | 11 | 11 | 10 | 9 | 12 | 12 | 11.7 |
| 50 | 12 | 15 | 23 | 13 | 23 | 19 | 24 | 19 | 18 | 20 | 18.8 |

**As it can be seen, any cut-off value, lower than 300, yields a bad performance, hence making it a bad choice.**

**Although the values of 500 and 1000 see good performance results, the minimum cut-off value is 300.**