CSC2023 Assignment 2 results

* CorrectnessTest: This class tests multiple normal cases and border cases

A screenshot of a computer

Description automatically generated

Where I test the first normal case that all three shapes must be on one shelf. Following after that I tested the second normal case that if a new shape is larger than current shelf height and it cannot be rotated and also there is still space on the sheet than new shelf must be created and place that shape. And the third test was if we added new shape and with border width of 300 than still, we can add it to the second shelf.

A screenshot of a computer

Description automatically generated

Finally the forth test I found an error and it was hard to track the fault where I added a new shape and in First fit algorithm it should be on first shelf and that was correct however, my loop in somehow did not want to rotate through all shelf to add the pervious shape in the second shelf. However, it worked perfectly in Next Fit as it creates new shelf to add that shape.

* PerformanceTest:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tests** | **First Fit** | **First Fit Sheet No** | **Next Fit** | **Next Fit** |
| **CPU time (Average)** | **(Average).** | **CPU time (Average).** | **Sheet No (Average).** |
| Test 1 | 5312 | 5105 | 32 | 4730 |
| Test 2 | 20591 | 10186 | 48 | 9411 |
| Test 3 | 50550 | 15270 | 71 | 14172 |
| Test 4 | 93613 | 20417 | 86 | 18952 |
| Test 5 | 151341 | 25519 | 114 | 23652 |
|  |  |  |  |  |

Comments on the results:

The above the table results showing that First Fit Algorithm was too slow compared to the Next Fit Algorithm as it was faster in performance. Since it will not look back on any decision that has been made, however, it is expected that First Fit to be slow as it should backtrack through all sheets and shelves that already exists which makes the computation longer. Nevertheless, First Fit used many sheets comparing to Next Fit and that I believe because there is a fault in the First Fit code. Actually, it was expected that First Fit would use fewer sheets. However, I made a fault in the code for First Fit, and it was hard to fix.

* SortedTest: Sorting was on height of the shape and in decreasing order.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tests** | **First Fit** | **First Fit Sheet No** | **Next Fit** | **Next Fit** |
| **CPU time (Average)** | **(Average).** | **CPU time (Average).** | **Sheet No (Average).** |
| Test 1 | 3931 | 4349 | 33 | 4377 |
| Test 2 | 12361 | 8737 | 67 | 8786 |
| Test 3 | 29667 | 12990 | 95 | 12800 |
| Test 4 | 62869 | 17336 | 124 | 17424 |
| Test 5 | 88082 | 21729 | 141 | 21833 |

Comments on the results:

For First Fit Algorithm we can say that we used First Fit Decreasing Algorithm and we can see from the table its performance has improved compare to it state without sorting, and it worked as expected use fewer sheets which means it gave an optimal number sheets and decreases the number of waste sheets and also, the CPU time was less compared to its state without sorting. Moreover, it has been found that the sheets used in Next Fit were a lot and that as we expected that Next Fit would not look back in any sheets. However, the performance was not fast as it was without sorting the shapes may be that because of the time taken to create new sheets as it used a large number of sheets.