Assignment 2: Glass Cutting Problem

170450661 – Kenneth Cajigas

Task 4: Correctness testing

To be able to verify that my algorithms have been implemented correctly, I will run a variety of tests. The first set of tests will be logic tests of my nextFit and firstFit methods are correct – that is:

- My nextFit method only considers the last sheet and shelf used to place a shape in
- My firstFit method considers every shelf in every sheet to place a shape in

The second set of tests will be rule checks that with a list of shapes they will verify if both my algorithms follow the rules stated in the spec have been met:

A. A shape is placed either at the bottom left corner of a sheet (starting the first shelf on the sheet) or to the right of another shape, if there is sufficient space in the shelf.

B. If a shape does not fit in a shelf, one rotates the shape and tries to fit it in the shelf.

C. If a shape still does not fit in a shelf, one can start a new shelf directly on top of the current shelf against the left side of the sheet, if there is enough space in the sheet. First, try to create a new shelf with the shape in its original orientation, and if it does not fit, one can rotate the shape and try to create the new shelf.

D. At most one shape is placed directly to the right of any other shape (see example 5 for clarification).

E. The total height of all shelves in a sheet does not exceed H

F. The number of shapes placed on a sheet cannot exceed L (where L is given).

Other considerations:

With my nextFit() algorithm, I'll also need to make sure that with the trace that I will draw by hand shows that when a shape can't be placed into a shelf, the shelf is considered "full" and when a shelf can't be placed to a sheet, the sheet will also be considered "full". In addition to this shape can't be added to any previous shelves or sheets.

With my firstFit() algorithm on the other hand, I will need to show in the trace and in the code output that shapes can be added to already existing shelves and shelves can be added to already existing sheets provided that there is space to add the shape in.

(In my trace drawings, if there is an R, this indicates that the shape was rotated so that it could be placed)

170450661 – Kenneth Cajigas

Testing the logic of my methods

Logic test 1 – Shapes

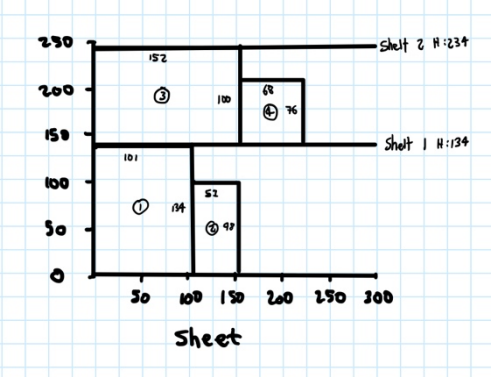
Purpose:
nextFit - Checking that shapes can be added into the last shelf used to place a shape
firstFit - Checking that shapes can be added into existing shelves if there is space

Number of shapes: 4 shapes within the height and width limit
Expected sheets: nextFit: 1 | firstFit: 2

Shape #	Height	Width
1	134	101
2	98	51

Shape #	Height	Width
3	100	152
4	76	68

nextFit - Running trace by hand:



It is expected that shape 4 should go into shelf 2 as nextFit considers shelf 1 to be "full" when it attempts to add shape 3 in as it can't fit.

Expected results

Sheet 1 (H234)	Shapes
Shelf 1 (H 134 W 152)	2
Shelf 2 (H 100 W 220)	2

Code output:

```
********** nextFit() testing *************

Used number of sheets for next fit: 1

***********************************************

Sheet 1 information

Sheet 1      - Total height: 234 | Shapes: 4         | Shelves: 2

Shelf 1      - Height used: 134  | Width: 152        | Shapes: 2
Shape 1      - Height: 134       | Width: 101
Shape 2      - Height: 98        | Width: 51

Shelf 2      - Height used: 100  | Width: 220        | Shapes: 2
Shape 1      - Height: 100       | Width: 152
Shape 2      - Height: 76        | Width: 68
```

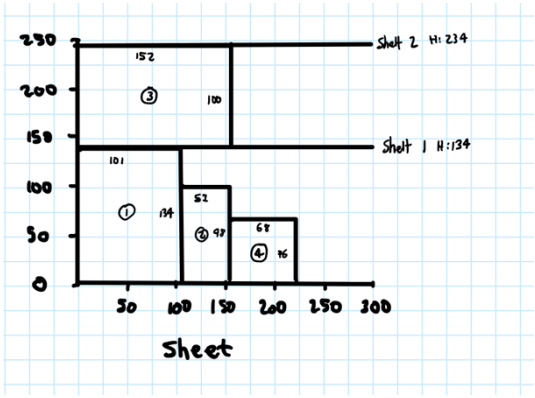
Test passed ✅

The results here match the trace I have drawn by hand.

As expected, shelves 1 and 2 have 2 shapes inside. Of them, with the expected shape widths and heights and every shape was added to the last shelf used to place a shape in.

170450661 – Kenneth Cajigas

firstFit - Running trace by hand:



Compared to nextFit, it is expected that shape 4 will be added to shelf 1 as it fits within the remaining height and width of the shelf.

This will show that my method does consider all existing shelves.

Expected results

Sheet 1 (H 234)	Shapes
Shelf 1 (H 134 W 220)	3
Shelf 2 (H 100 W 152)	1

Code output:

```
********** firstFit() testing **********

Used number of sheets for first fit: 1

************************************************

Sheet 1 information

Sheet 1      - Total height: 234 | Shapes: 4         | Shelves: 2

Shelf 1      - Height used: 134  | Width: 220        | Shapes: 3
Shape 1      - Height: 134       | Width: 101
Shape 2      - Height: 98        | Width: 51
Shape 3      - Height: 76        | Width: 68

Shelf 2      - Height used: 100  | Width: 152        | Shapes: 1
Shape 1      - Height: 100       | Width: 152
```

Test passed ✅

As expected, the output does match the trace that I have drawn above.

Most importantly, shelf 1 has 3 shapes inside it, with Shape 4 (H76 W 68) being correctly added into shelf 1 of the sheet and shelf 2 only having 1 shape.

170450661 – Kenneth Cajigas

Logic test 2 – Shelves

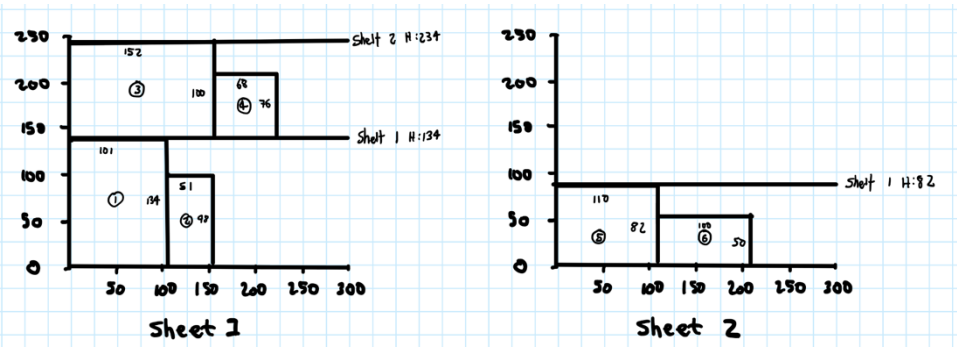
Purpose:
nextFit - Checking that any previously considered "full" shelves don't have shapes added to them
firstFit – Checking that all existing shelves are considered when adding a shape.

Number of shapes: 6 shapes within the height and width limit
Expected sheets: nextFit: 2 | firstFit: 1

Shape #	Height	Width
1	134	101
2	98	51
3	100	152

Shape #	Height	Width
4	76	68
5	82	110
6	50	100

nextFit - Running trace by hand:



As nextFit only works from the last used shelf and sheet, it is expected shape 5 will be attempted to be added to shelf 2 of sheet 1. It will fail and shelf 2 of sheet 1 will be considered "full". Sheet 1 will also be considered full, so a new sheet with the shelf will be created. Shape 6 should also be added to the new shelf in sheet 2.

Expected results

Sheet 1 (H 234)	Shapes
Shelf 1 (H 134 W 152)	2
Shelf 2 (H 100 W 220)	2

Sheet 2 (H 82)	Shapes
Shelf 1 (H 82 W 210)	2

Code output:

```
********** nextFit() testing ************

Used number of sheets for next fit: 2

*********************************************

Sheet 1 information

Sheet 1      - Total height: 234 | Shapes: 4         | Shelves: 2

Shelf 1      - Height used: 134  | Width: 152        | Shapes: 2
Shape 1      - Height: 134       | Width: 101
Shape 2      - Height: 98        | Width: 51

Shelf 2      - Height used: 100  | Width: 220        | Shapes: 2
Shape 1      - Height: 100       | Width: 152
Shape 2      - Height: 76        | Width: 68

*********************************************

Sheet 2 information

Sheet 2      - Total height: 82  | Shapes: 2         | Shelves: 1

Shelf 1      - Height used: 82   | Width: 210        | Shapes: 2
Shape 1      - Height: 82        | Width: 110
Shape 2      - Height: 50        | Width: 100
```

Test passed ✅

The results here match the trace I have drawn by hand.

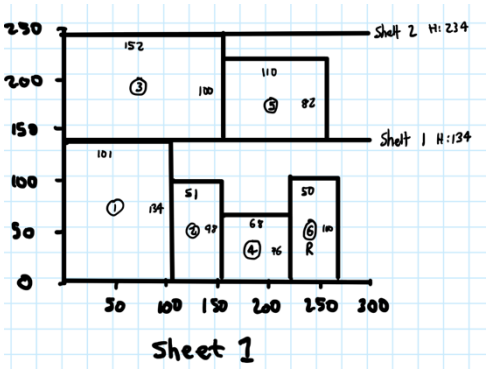
As expected, when Shape 5 (H 82 W 110) was attempted to be added to shelf 2, this was considered full, so a new shelf had to be made.

However, in addition to this, there was no space so sheet 1 was also considered full and a new sheet was made.

Shape 6 (H 50 W 100) was also added to the new shelf in sheet 2.

170450661 – Kenneth Cajigas

firstFit - Running trace by hand:



Compared to the same test run in nextFit, firstFit will not need another sheet to be made here. This is because my method should consider every existing shelf in sheet 1, allowing Shapes 5 and 6 to be added to sheet 1.

Expected results

Sheet 1 (H 234)	Shapes
Shelf 1 (H 134 W 270)	4
Shelf 2 (H 100 W 262)	2

Code output:

```
********** firstFit() testing **********

Used number of sheets for first fit: 1

************************************************

Sheet 1 information

Sheet 1      - Total height: 234 | Shapes: 6         | Shelves: 2

Shelf 1      - Height used: 134  | Width: 270        | Shapes: 4
Shape 1      - Height: 134       | Width: 101
Shape 2      - Height: 98        | Width: 51
Shape 3      - Height: 76        | Width: 68
Shape 4      - Height: 100       | Width: 50

Shelf 2      - Height used: 100  | Width: 262        | Shapes: 2
Shape 1      - Height: 100       | Width: 152
Shape 2      - Height: 82        | Width: 110
```

Test passed ✅

The results here match the trace I have drawn by hand and no other sheets were created.

Shape 5 (H 82 W 110) has correctly been added to shelf 2 of sheet 1

Shape 6 (H 50 W 100) has correctly been added to shelf 1 of sheet 1

170450661 – Kenneth Cajigas

Logic test 3 – Sheets

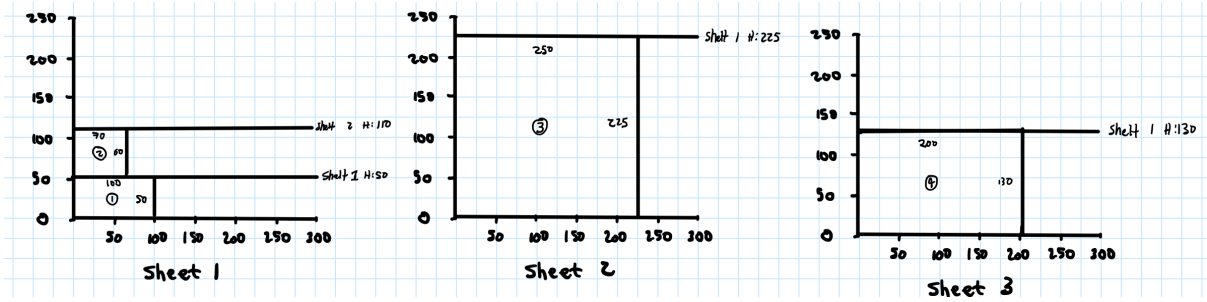
Purpose:
nextFit - Checking that only the last used sheet is considered when adding a new shelf to it
firstFit - Checking that all existing sheets are considered when adding in a shelf

Number of shapes: 4 shapes within the height and width limit
Expected sheets: nextFit: 3 | firstFit: 2

Shape #	Height	Width
1	50	100
2	60	70

Shape #	Height	Width
3	225	250
4	130	200

nextFit - Running trace by hand:



When adding in shape 4, it can't be added to the last shelf in sheet 2 so a new shelf is created. The shelf can't fit into sheet 2, so a new sheet must be created.

Expected results

Sheet 1 (H 110)	Shapes
Shelf 1 (H 50 W 100)	1
Shelf 2 (H 60 W 70)	1

Sheet 2 (H 225)	Shapes
Shelf 1 (H 225 W 250)	1

Sheet 3 (130)	Shapes
Shelf 1 (H 130 W 200)	1

Code output:

```
********** nextFit() testing *************

Used number of sheets for next fit: 3

***********************************************

Sheet 1 information

Sheet 1      - Total height: 110 | Shapes: 2         | Shelves: 2

Shelf 1      - Height used: 50   | Width: 100        | Shapes: 1
Shape 1      - Height: 50        | Width: 100

Shelf 2      - Height used: 60   | Width: 70         | Shapes: 1
Shape 1      - Height: 60        | Width: 70

***********************************************

Sheet 2 information

Sheet 2      - Total height: 225 | Shapes: 1         | Shelves: 1

Shelf 1      - Height used: 225  | Width: 250        | Shapes: 1
Shape 1      - Height: 225       | Width: 250

***********************************************

Sheet 3 information

Sheet 3      - Total height: 130 | Shapes: 1         | Shelves: 1

Shelf 1      - Height used: 130  | Width: 200        | Shapes: 1
Shape 1      - Height: 130       | Width: 200
```

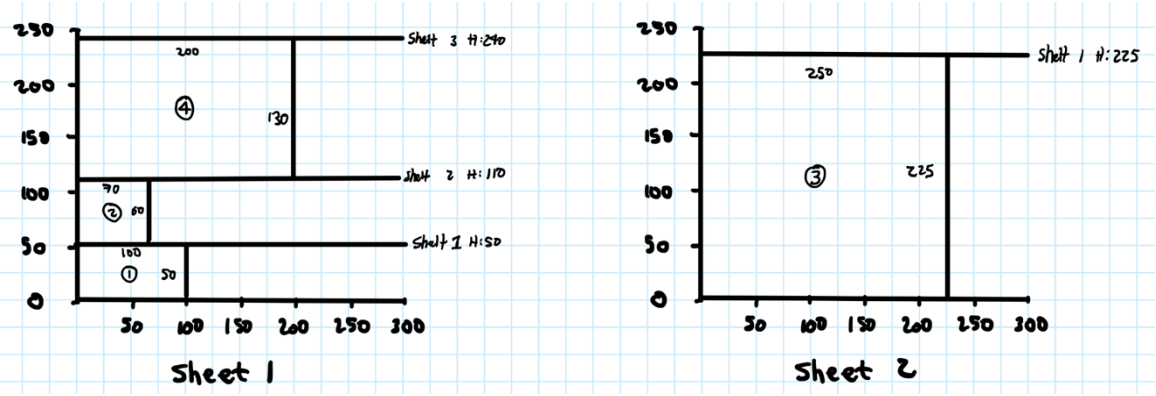
Test passed ✅

The results here match the trace I have drawn by hand.

When attempting to add Shape 3 (H 225 W 250) to sheet 1, there was no space in the shelves, so a new shelf was created, but there was no space for the new shelf. As a result, a new sheet was created as expected.

This was also the same for adding Shape 4 (H 110 W 200). It could not fit into shelf 1 of sheet 2, so a new shelf was created which couldn't fit into sheet 2, so a new sheet was created.

firstFit - Running trace by hand:



Compared to the same test run in nextFit, there should be one less sheet created. This is when shape 4 can't be added into any existing shelf in every sheet, a new shelf is created. My algorithm should go back to the first sheet and see if there is any space to fit in the new shelf. The new shelf with shape 4 will be able to fit into sheet 1.

Expected results

Sheet 1 (H 240)	Shapes
Shelf 1 (H 50 W 100)	1
Shelf 2 (H 60 W 70)	1
Shelf 3 (H 130 W 200)	1

Sheet 1 (H 225)	Shapes
Shelf 1 (H 225 W 250)	1

Code output:

```
********** firstFit() testing *************

Used number of sheets for first fit: 2

*********************************************

Sheet 1 information

Sheet 1      - Total height: 240 | Shapes: 3         | Shelves: 3

Shelf 1      - Height used: 50   | Width: 100        | Shapes: 1
Shape 1      - Height: 50        | Width: 100

Shelf 2      - Height used: 60   | Width: 70         | Shapes: 1
Shape 1      - Height: 60        | Width: 70

Shelf 3      - Height used: 130  | Width: 200        | Shapes: 1
Shape 1      - Height: 130       | Width: 200

*********************************************

Sheet 2 information

Sheet 2      - Total height: 225 | Shapes: 1         | Shelves: 1

Shelf 1      - Height used: 225  | Width: 250        | Shapes: 1
Shape 1      - Height: 225       | Width: 250
```

Test passed ✅

The results here match the trace I have drawn by hand.

The new shelf that had shape 4 in it was correctly added into sheet 1.

170450661 – Kenneth Cajigas

Testing against rules A-F

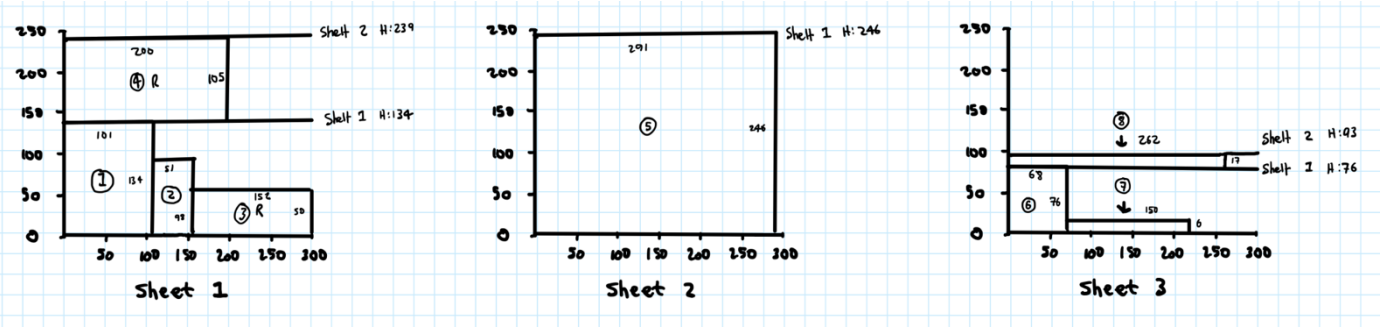
Rule test 1 – Testing against rules A-E

Purpose: Checking that shapes can be added in correctly with new shelves and sheets created
Number of shapes: 8 shapes within the height and width limit
Expected sheets: nextFit: 3 | firstFit: 3

Shape #	Height	Width
1	134	101
2	98	51
3	146	50
4	200	105

Shape #	Height	Width
5	246	291
6	76	68
7	6	150
8	17	262

nextFit - Running trace by hand:



Code output and rules met:

My code passes this test as each sheet matches the trace drawn by hand along with the rules below.

```
********** nextFit() testing **********

Used number of sheets for next fit: 3

***********************************************

Sheet 1 information

Sheet 1      - Total height: 239 | Shapes: 4         | Shelves: 2

Shelf 1      - Height used: 134  | Width: 298        | Shapes: 3
Shape 1      - Height: 134       | Width: 101
Shape 2      - Height: 98        | Width: 51
Shape 3      - Height: 50        | Width: 146

Shelf 2      - Height used: 105  | Width: 105        | Shapes: 1
Shape 1      - Height: 105       | Width: 200

***********************************************

Sheet 2 information

Sheet 2      - Total height: 246 | Shapes: 1         | Shelves: 1

Shelf 1      - Height used: 246  | Width: 291        | Shapes: 1
Shape 1      - Height: 246       | Width: 291

***********************************************

Sheet 3 information

Sheet 3      - Total height: 93  | Shapes: 3         | Shelves: 2

Shelf 1      - Height used: 76   | Width: 218        | Shapes: 2
Shape 1      - Height: 76        | Width: 68
Shape 2      - Height: 6         | Width: 150

Shelf 2      - Height used: 17   | Width: 262        | Shapes: 1
Shape 1      - Height: 17        | Width: 262
```

Rule A:
Shapes 1, 4, 5 and 6 has been correctly placed on the bottom left corner of their shelves.

Rule B:
Shape 3 in its original orientation couldn't fit into shelf 1 of sheet 1, but rotating it does allow it to fit in.

Rule C:
Shelf 2 of sheet 1 with shape 4 would not have fitted with its original orientation, rotating the shape allowed this.

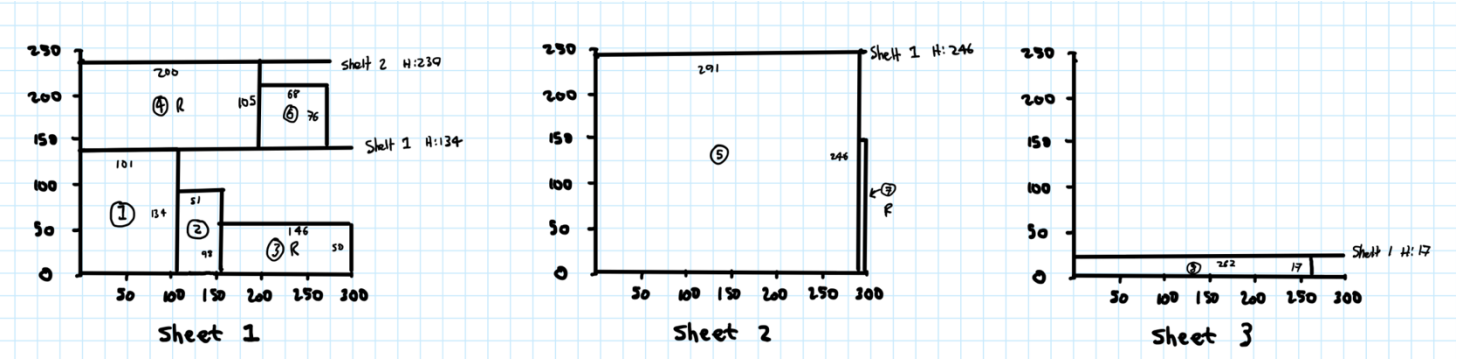
Shape 8 couldn't fit in shelf 1 of sheet 3 and so a new shelf was created on top of shelf 1 and with the shape in its original orientation

Rule D:
See shapes 2, 3 and 7 – they are all placed to the right of another shape

Rule E:
The total height of all shelves in all sheets from this list of shapes does not exceed H which is currently set to 250.

170450661 – Kenneth Cajigas

firstFit – Running trace by hand:



With nextFit, the difference here is that Shape 6 gets added to shelf 2 of sheet 1 and Shape 7 gets added to shelf 1 of sheet 2 after being rotated as the algorithm considers all existing shelves in every sheet. These shelves can accommodate these shapes.

Code output and rules met:

```
********** firstFit() testing *************

Used number of sheets for first fit: 3

*********************************************

Sheet 1 information

Sheet 1      - Total height: 239 | Shapes: 5         | Shelves: 2

Shelf 1      - Height used: 134  | Width: 298        | Shapes: 3
Shape 1      - Height: 134       | Width: 101
Shape 2      - Height: 98        | Width: 51
Shape 3      - Height: 50        | Width: 146

Shelf 2      - Height used: 105  | Width: 268        | Shapes: 2
Shape 1      - Height: 105       | Width: 200
Shape 2      - Height: 76        | Width: 68

*********************************************

Sheet 2 information

Sheet 2      - Total height: 246 | Shapes: 2         | Shelves: 1

Shelf 1      - Height used: 246  | Width: 297        | Shapes: 2
Shape 1      - Height: 246       | Width: 291
Shape 2      - Height: 150       | Width: 6

*********************************************

Sheet 3 information

Sheet 3      - Total height: 17  | Shapes: 1         | Shelves: 1

Shelf 1      - Height used: 17   | Width: 262        | Shapes: 1
Shape 1      - Height: 17        | Width: 262
```

My code passes this test as each sheet matches the trace drawn by hand along with the rules below:

Rule A:
Shapes 1, 4, 5 and 8 has been correctly placed on the bottom left corner of their shelves

Rule B:
Shape 3 in its original orientation couldn't fit into shelf 1 of sheet 1, but rotating it does allow it to fit in.

Shape 7 did not fit shelf 1 of sheet 2 in its original orientation, however rotating it does allow it to fit in the shelf.

Rule C:
Shelf 2 of sheet 1 with shape 4 would not have fitted with its original orientation, rotating the shape allowed this.

Rule D:
Shapes 2, 3, 6 and 7 are all placed to the right of another shape

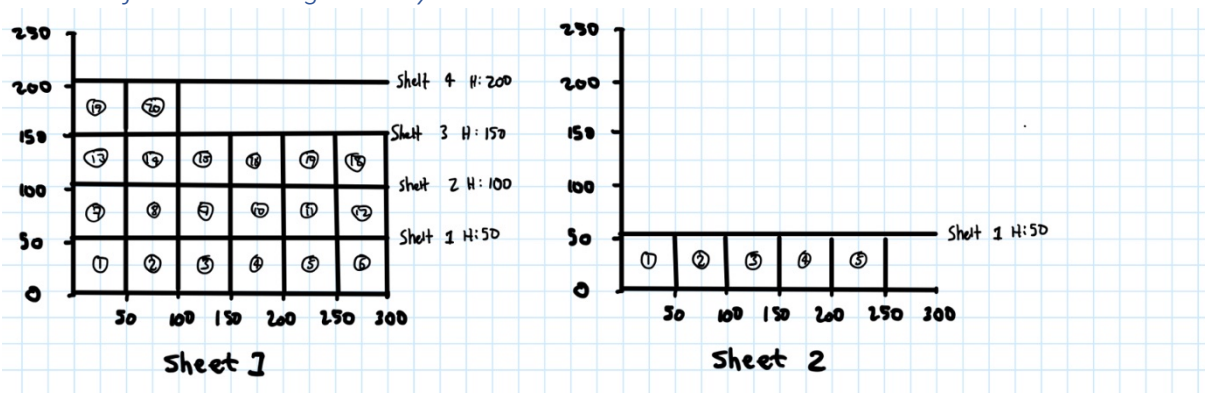
Rule E:
The total height of all shelves in all sheets from this list of shapes does not exceed H which is currently set to 250.

170450661 – Kenneth Cajigas

Rule test 2 – Checking rule F

Purpose: Making sure that when a sheet reaches the max limit (L) that a new sheet is created
Number of shapes: 25 shapes of height and width of 50
Expected sheets: nextFit: 2 | firstFit: 2

nextFit & firstFit Running trace by hand:



Code output:

```
********** nextFit() testing *************

Used number of sheets for next fit: 2

************************************************

Sheet 1 information

Sheet 1      - Total height: 200 | Shapes: 20        | Shelves: 4

Shelf 1      - Height used: 50   | Width: 300        | Shapes: 6
Shape 1      - Height: 50        | Width: 50
Shape 2      - Height: 50        | Width: 50
Shape 3      - Height: 50        | Width: 50
Shape 4      - Height: 50        | Width: 50
Shape 5      - Height: 50        | Width: 50
Shape 6      - Height: 50        | Width: 50

Shelf 2      - Height used: 50   | Width: 300        | Shapes: 6
Shape 1      - Height: 50        | Width: 50
Shape 2      - Height: 50        | Width: 50
Shape 3      - Height: 50        | Width: 50
Shape 4      - Height: 50        | Width: 50
Shape 5      - Height: 50        | Width: 50
Shape 6      - Height: 50        | Width: 50

Shelf 3      - Height used: 50   | Width: 300        | Shapes: 6
Shape 1      - Height: 50        | Width: 50
Shape 2      - Height: 50        | Width: 50
Shape 3      - Height: 50        | Width: 50
Shape 4      - Height: 50        | Width: 50
Shape 5      - Height: 50        | Width: 50
Shape 6      - Height: 50        | Width: 50

Shelf 4      - Height used: 50   | Width: 100        | Shapes: 2
Shape 1      - Height: 50        | Width: 50
Shape 2      - Height: 50        | Width: 50

************************************************

Sheet 2 information

Sheet 2      - Total height: 50  | Shapes: 5         | Shelves: 1

Shelf 1      - Height used: 50   | Width: 250        | Shapes: 5
Shape 1      - Height: 50        | Width: 50
Shape 2      - Height: 50        | Width: 50
Shape 3      - Height: 50        | Width: 50
Shape 4      - Height: 50        | Width: 50
Shape 5      - Height: 50        | Width: 50
```

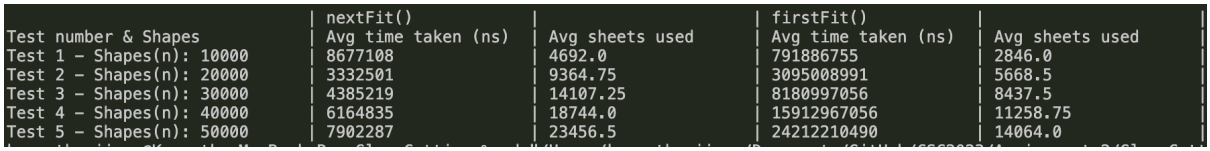
```
********** firstFit() testing *************

Used number of sheets for first fit: 2

************************************************

Sheet 1 information

Sheet 1      - Total height: 200 | Shapes: 20        | Shelves: 4

Shelf 1      - Height used: 50   | Width: 300        | Shapes: 6
Shape 1      - Height: 50        | Width: 50
Shape 2      - Height: 50        | Width: 50
Shape 3      - Height: 50        | Width: 50
Shape 4      - Height: 50        | Width: 50
Shape 5      - Height: 50        | Width: 50
Shape 6      - Height: 50        | Width: 50

Shelf 2      - Height used: 50   | Width: 300        | Shapes: 6
Shape 1      - Height: 50        | Width: 50
Shape 2      - Height: 50        | Width: 50
Shape 3      - Height: 50        | Width: 50
Shape 4      - Height: 50        | Width: 50
Shape 5      - Height: 50        | Width: 50
Shape 6      - Height: 50        | Width: 50

Shelf 3      - Height used: 50   | Width: 300        | Shapes: 6
Shape 1      - Height: 50        | Width: 50
Shape 2      - Height: 50        | Width: 50
Shape 3      - Height: 50        | Width: 50
Shape 4      - Height: 50        | Width: 50
Shape 5      - Height: 50        | Width: 50
Shape 6      - Height: 50        | Width: 50

Shelf 4      - Height used: 50   | Width: 100        | Shapes: 2
Shape 1      - Height: 50        | Width: 50
Shape 2      - Height: 50        | Width: 50

************************************************

Sheet 2 information

Sheet 2      - Total height: 50  | Shapes: 5         | Shelves: 1

Shelf 1      - Height used: 50   | Width: 250        | Shapes: 5
Shape 1      - Height: 50        | Width: 50
Shape 2      - Height: 50        | Width: 50
Shape 3      - Height: 50        | Width: 50
Shape 4      - Height: 50        | Width: 50
Shape 5      - Height: 50        | Width: 50
```

Both algorithms have passed the test ✅

Task 5: Performance testing

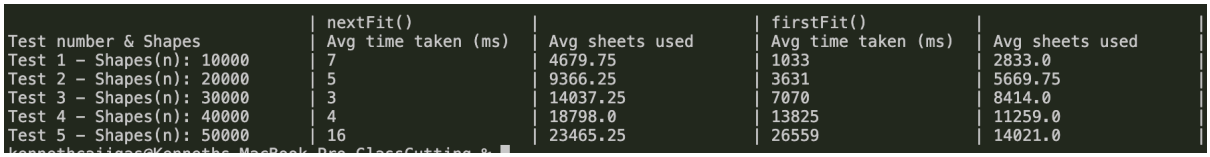
Code output of results

Nanoseconds



	nextFit()		firstFit()	
Test number & Shapes	Avg time taken (ns)	Avg sheets used	Avg time taken (ns)	Avg sheets used
Test 1 – Shapes(n): 10000	8677108	4692.0	791886755	2846.0
Test 2 – Shapes(n): 20000	3332501	9364.75	3095008991	5668.5
Test 3 – Shapes(n): 30000	4385219	14107.25	8180997056	8437.5
Test 4 – Shapes(n): 40000	6164835	18744.0	15912967056	11258.75
Test 5 – Shapes(n): 50000	7902287	23456.5	24212210490	14064.0

Milliseconds



	nextFit()		firstFit()	
Test number & Shapes	Avg time taken (ms)	Avg sheets used	Avg time taken (ms)	Avg sheets used
Test 1 – Shapes(n): 10000	7	4679.75	1033	2833.0
Test 2 – Shapes(n): 20000	5	9366.25	3631	5669.75
Test 3 – Shapes(n): 30000	3	14037.25	7070	8414.0
Test 4 – Shapes(n): 40000	4	18798.0	13825	11259.0
Test 5 – Shapes(n): 50000	16	23465.25	26559	14021.0

Analysis

The above test results are from running 5 different tests and repeating each test 4 times.

Time taken

From the results above in an overview the most efficient algorithm based on the shortest time taken to complete is nextFit.

This is expected because with nextFit, there is no need for the algorithm to have to analyse every shelf in a sheet. Once a shelf is considered full up, it simply creates a new one, tries to place it in the current sheet, and if that sheet is full up, it will create a new one and move on to the next shape.

With firstFit, it takes the longest because it has to go through every single shelf within each sheet used to see if there is a fit for a shape, and if there isn't then it will create a new shelf and then try to go through each sheet again to fit in the shelf. If this still fails, at this point it will create a new sheet and add that shelf to it.

Sheets used

On the other hand, the most efficient algorithm based on used sheets used is firstFit. As the nature of firstFit involves the algorithm considering every shelf in each sheet, it means that space is used more efficiently as shapes are placed in the next available space.

NextFit uses sheets less efficiently, because a shelf is considered "full" when a shape can't fit inside, and a sheet is considered "full" because a shelf can't fit inside it, regardless of if there is still lots of space available.

170450661 – Kenneth Cajigas

Task 6: Sorted shapes

Overview

I have decided to sort the shapes based on 6 conditions:
(Increasing first puts the smallest shape first and vice versa)

Height, Width and Area – (All sorted by increased and decreasing order)

	Sorted by (avg sheets used):						
Algorithm	Unsorted	Increasing height	Decreasing height	Increasing width	Decreasing width	Increasing area	Decreasing area
Shapes: 10000 – nextFit()	4647.0	4251.25	4191.25	4547.0	4514.25	4580.25	4605.75
Shapes: 10000 – firstFit()	2827.5	3619.0	2604.0	3432.25	2690.75	3939.0	2627.25
Shapes: 20000 – nextFit()	9366.5	8467.0	8437.25	9048.5	9014.25	9150.0	9114.25
Shapes: 20000 – firstFit()	5642.0	7108.0	5219.0	6833.0	5324.0	7924.0	5174.75
Shapes: 30000 – nextFit()	14094.75	12764.5	12715.25	13557.0	13615.0	13641.0	13577.5
Shapes: 30000 – firstFit()	8445.0	10637.0	7850.75	10188.25	8006.75	11863.0	7719.75
Shapes: 40000 – nextFit()	18636.25	17004.0	16874.5	18026.25	18062.0	18089.5	18121.75
Shapes: 40000 – firstFit()	11137.25	14112.0	10377.5	13599.5	10599.5	15751.0	10355.75
Shapes: 50000 – nextFit()	23457.75	21161.0	21149.75	22595.25	22569.25	22628.75	22559.0
Shapes: 50000 – firstFit()	14037.25	17479.25	12978.75	17008.0	13169.25	19763.0	12920.75

The results above are from 5 different tests, with each test being repeated 4 times for accurate results. In each test, the unsorted shape between nextFit and firstFit was (1.64%, 1.66%, 1.66%, 1.67%, 1.67. Averaging at 1.66% difference)

nextFit() sorting analysis

Looking at the results above, it is most beneficial to sort the shapes by decreasing order of height, where the largest height shape is put in first. This was the case across all 5 tests.

The nature of nextFit() where it places shapes where next available means that space is used as efficiently as possible because for every shape, there is never going to be a shape that is larger than the first shape placed in a shelf. Meaning that regardless of the width taken by a Shape, a shelf will never be considered full too early because a shape's height exceeds the shelf's height. By doing this, it means that every shelf is used up as much as possible with the optimal number of shapes inside it.

firstFit() sorting analysis

Looking at the results above, it is most beneficial to sort the shapes by decreasing order of area, where the shape with the largest area is put in first. Sorting by decreasing height had similar results. Overall, sorting by decreasing area was quickest each time. This is because with firstFit considering every shelf in every sheet, it is optimal to place the largest area of shapes first, as down the line shapes with the smallest area will be slotted into any spaces in shelves available.

Overall analysis and recommended algorithm

If the manufacturer would like to reduce the number of sheets of glass used, the best algorithm to use overall would be to use firstFit. Based on the results above, sorting the shapes by decreasing order of area, compared to not sorting the shapes in firstFit, this has decreased the number of sheets used by 1.08% across every test on average against not sorting the shapes. Comparing the optimal number of sheets produced by nextFit and the optimal number of sheets produced by firstFit, on average firstFit produces 1.62% fewer sheets.

If they are happy to have a compromise between the number of sheets produced and speed, then nextFit would still be a viable option, as long as shapes are sorted by decreasing height.