3

(a)

The recommended sorting algorithm for the library would be Mergesort. This is because the libraries data must remain in the same order of any duplicate books, and due to only local swaps within Mergesort means the selected algorithm is stable. Since Mergesort also has guaranteed $\Theta(n\log(n))$ time in the worst case the algorithm is also fast for all amounts of books.

(b)

The recommended sorting algorithm for the manufacturer would be Selection sort. Selection sort has very high comparisons as it must identify the smallest value left in the non-sorted section of the array, but will always have $\Theta(n)$ swaps, given after n smallest values the array is fully sorted. This low swapping high comparisons algorithm is ideal for this scenario, since its slow speed is irrelevant.

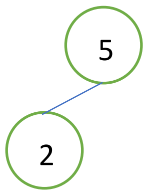
(c)

The recommended sorting algorithm for the observatory would be Insertion sort. Given only 50 values need to be sorted, insertion sort will perform faster at this level than most of the more complex algorithms which also require more memory. The selected algorithm only requires constant memory which is part of the specification.

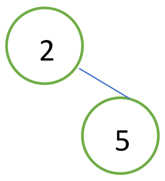
4

For the BST to always be a stick with nodes only existing in the right pointers of the tree, the nodes must be rotated if they are placed in the left pointer. For instance if the BST is initialised with 5 as the roots value the tree looks like this.

Then let 2 be inserted, 2 is less than 5 and thus not in the right-hand pointer which is required for a stick.



And as such the Stickify() function will perform a right rotation on the root and pivot/new. This will cause the left-hand pointer of the root to become NULL, and the new nodes right-hand pointer to become the root. The root maintains its right-hand pointer and if the root had a root, this parents (assumed) right pointer would now point to the new node.



In the case that the new node's value is greater than that of the root and is placed in the right pointer, no further action needs be taken given the stick flows in that direction.

```
Stickify(new, root){
        if(new.value < root.value){
                RotateRight(node)
        }
}
```