

COMP20007 Assignment 2

Zi Ying Ng (Joanne)

1085130

Question 3

- (a) Mergesort, because it is stable thus maintaining the original order, and it is fast with guaranteed $O(n \log n)$ time complexity.
- (b) Selection Sort, because finding the next element is efficient, and each swap guarantees an item is in the right position. Even though it is $O(n^2)$, the number of elements is small enough to be feasible.
- (c) Counting Sort, because it only requires an auxiliary array of size 11 due to the range of values, while sorting takes place in $O(n + k)$ for $n = 50$ and $k = 11$, which is constant time and fast enough to occur in a millisecond assuming around 10^9 operations per second.

Question 4

```
function STICKIFY(new, root)
    if new.value < root.value then
        ROTATERIGHT(root)
```

Assuming an empty structure, the first insertion only has one element. If the new node is larger than the first, it will be inserted into the right node, so the BST is a stick; otherwise, it will be inserted into the left node, and the stickify algorithm will rotate the root node right, converting the BST into a stick. Hence, the BST always starts off as a stick.

Assuming we have a stick, if the new value is smaller than the root, it will be inserted into the left node of the root and rotated right once; otherwise, BSTInsert will be called on the right node of the root, which becomes the root node. Hence, the new node is either inserted into the left node of the first node it is smaller than and that root node is rotated right once to form a stick, or the new node will be inserted into the right node of the last node if it is larger than all the nodes.

Since we always start with a stick, and will always have a stick given a stick, the algorithm works.