**Question 3**

**Problem 1:** Merge Sort, because it is stable (maintains the original order of duplicate books) and has guaranteed worst case performance, which makes it fast enough to inform the librarian before end of day.

**Problem 2:** Selection Sort - it is insensitive to input data, best/average/worst case performance are all , which is good for sorting arbitrary records. Also, each item is swapped directly into their correct position, hence minimising record swaps by requiring a maximum of swaps.

**Solution 3:** Counting Sort, because the input is a set of non-negative integers and the algorithm has a linear runtime O(n+k), with k = 10 in this case, as its worst case performance. As the maximum key value (10) is significantly smaller than the number of values to be sorted, Counting Sort is very space efficient as it only uses an array of space O(10) to store its counts.

**Question 4**

function Stickify(new, root):

if root.left != NULL then

RotateRight(root)

**Diagram + Explanation**

A close up of text on a white background

Description automatically generated