Problem 3

a) **Merge Sort:**
For its stability, it can maintain the original order. On top of that, its operation is in O(n), which means only one merge operation can sort the last box (sorted) and all books before that (also sorted).

b) **Selection Sort:**
For record swapping is much more energy consuming than record comparison and we need to work without over n swaps that is less than other sorting algorithms, Selection Sort meets the needs.

c) **Counting Sort:**
Known the range of discrete values, Counting Sort has O(n) performance. To use the space that is less than luminosity values, counting sort needs only O(r) additional space, which meets the requirement.

Problem 4

```
function Stickify(new, root):
    if new = root.left then
        ROTATERIGHT(root)
        new.right <- root
    return
```