CPTS350 HW4

Tristan Miller
2024-02-05 6:30 PM
Links: WSU CPTS

1. On sorting purple-haired babies...

a.

The simplest way to manage the sorting of these babies in one-pass and in-place is to line the babies up single file (array) with the babies of varied hair colors in random places throughout the line. We can have two adults (pointers) start together at the beginning of the line. Adult 1 will traverse the line until they reach the first baby with brown hair, while Adult 2 will traverse the line staying at the first known baby with purple hair. Once they have each found their baby, they will swap the babies in the line. Once Adult 1 is at the last baby in line, and Adult 2 is at the last purple-haired baby in the line, the sorting is complete. In this sense, it only took one pass of the line to sort all the babies, and the time it took is linear, since it depends solely on how many babies were in the line.

b.

In this case, I consider each of the requirements in order: "one-pass" means the adults should not have to backtrack through the line. "In-place" means there should be no need to put the babies anywhere besides the line that already exists. "Linear time" means that the only factor that should make this take longer is the addition of more babies, and the addition of each baby should add the same amount of time to the sorting.

c.

```
array babySorter9000(arr) {
        // Place Adult 1 & Adult 2 at the beginning of the line:
        for (int i=0, j=0; i < arr.length; i++) {
                // Adult 1 traverses until they find brown hair:
                while (arr[i] ≠ brownHair)
                        i++;
                // Adult 2 traverses until they are at the first baby with purple hair:
                while (arr[j] ≠ purpleHair)
                        j++;
                // Adult 1 should be further in the line
                if (i > j)
                        swap(arr[i], arr[j]);
        }
        return the-sorted-array;
}
```

2.

This is totally doable if we get to add just one more pointer:

```
array babySorter9001(arr) {
        // Place Adult 1 & Adult 2 at the beginning of the line:
        for (int i=0, j=0, k=0; i < arr.length; i++) {
                // Adult 1 traverses until they find brown hair:
                while (arr[i] != brownHair)
                        i++;
                // Adult 2 traverses until they are at the first baby with purple hair:
                while (arr[j] != purpleHair)
                        j++;
                // Adult 3 traverses until they are at the first baby with black hair:
                while (arr[k] != blackHair)
                        k++;
                // Check whether Adult 2 or Adult 3 are further back in line:
                if (j < k) {
                        // Adult 1 should be further in the line and swap with Adult 2
                        if (i > j)
                                swap(arr[i], arr[j]);
                }
                else {
                        // Adult 1 should be further in the line and swap with Adult 3
                        if (i > k)
                                swap(arr[i], arr[k]);
                }
        }
        return the-sorted-array;
}
```