35.1 Counting Sort

Imagine if instead of driving a slow Honda Civic, we started driving a fast Ferrari. Unfortunately, we won't actually be driving in a Ferrari today, but we will witness a blazing fast algorithm that's just as fast called Radix Sorts.

When sorting an array, sorting requires $\Omega(N \log N)$compare operations in the worst case (array is sorted in descending order). Thus, the ultimate comparison based sorting algorithm has a worst case runtime of $\Theta(N \log N)$.

From an asymptotic perspective, that means no matter how clever we are, we can never beat Merge Sort’s worst case runtime of $\Theta(N \log N)$. But what if we don't compare at all?

#			
5	Sandra	Vanilla	Grimes
0	Lauren	Mint	Jon Talabot
11	Lisa	Vanilla	Blue Peter
9	Dave	Chocolate	Superpope
4	JS	Fish	The Filthy Reds
7	James	Rocky Road	Robots are Supreme
3	Edith	Vanilla	My Bloody Valentine
6	Swimp	Chocolate	Sef
1	Delbert	Strawberry	Ronald Jenkees
2	Glaser	Cardamom	Rx Nightly
8	Lee	Vanilla	La(r)va
10	Bearman	Butter Pecan	Extrobophile

#			
0	Lauren	Mint	Jon Talabot
1	Delbert	Strawberry	Ronald Jenkees
2	Glaser	Cardamom	Rx Nightly
3	Edith	Vanilla	My Bloody Valentine
4	JS	Fish	The Filthy Reds
5	Sandra	Vanilla	Grimes
6	Swimp	Chocolate	Sef
7	James	Rocky Road	Robots are Supreme
8	Lee	Vanilla	La(r)va
9	Dave	Chocolate	Superpope
10	Bearman	Butter Pecan	Extrobophile
11	Lisa	Vanilla	Blue Peter

Left is original, right is ordered output

Essentially what just happened is that we first made a new array of the same size and then just copied all of the # indexes to the correct location. So first we look at 5 Sandra Vanilla Grimes and then copy this over to the 5th index in our new array.

This does guarantee $\Theta(N)$ worst case time. However what if we were working with

- Non-unique keys.
- Non-consecutive keys.
- Non-numerical keys.

All of these cases are complex cases that aren't so simple to deal with. Essentially what we can do is create a simpler method which is to:

- Count number of occurrences of each item.
- Iterate through list, using count array to decide where to put everything.

Bottom line, we can use counting sort to sort N objects in $\Theta(N)$ time.

Previous
35. Radix Sorts

Next
35.2 LSD Radix Sort

Last updated 1 year ago