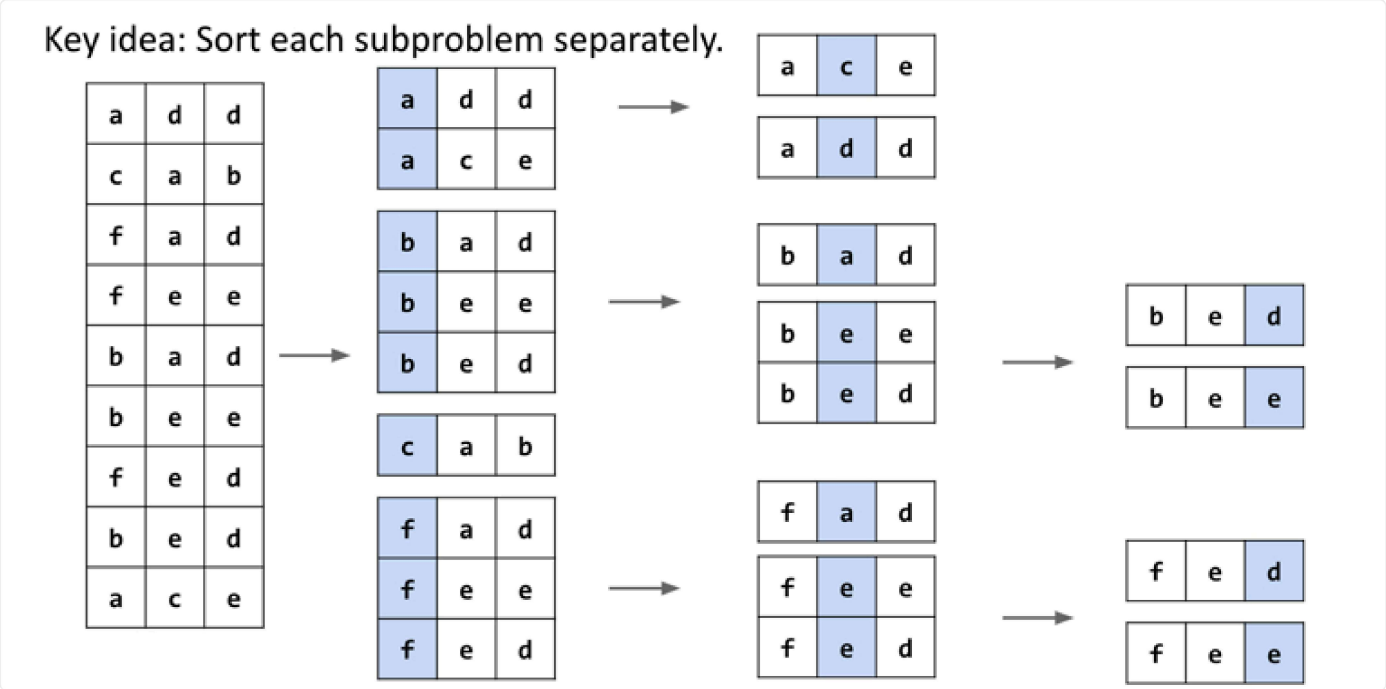
35.3 MSD Radix Sort

Basic idea: Just like LSD, but sort from leftmost digit towards the right.

> Suppose we sort by topmost digit, then middle digit, then rightmost digit. Will we arrive at the correct result?



MSD Radix Sort (correct edition)

Notice first we sorted by leftmost digit. Then we grouped the data by the leftmost digit, so one group would start with a's, then the next group would start with b's, and so on and so forth. Then within our subgroups we would order by middle digit, and create newer subgroups. And finally we would break this up into further subgroups until we have all individual subproblems. This final result would be sorted.

Runtimes

Best Case.

- We finish in one counting sort pass, looking only at the top digit: $\Theta(N + R)$

Worst Case.

- We have to look at every character, degenerating to LSD sort: $\Theta(WN + WR)$

Summary of Runtimes

	Memory	Runtime (worst)	Notes	Stable?
Heapsort	Θ(1)	Θ(N log N)*	Bad caching (61C)	No
Insertion	Θ(1)	$\Theta(N^2)$*	Fastest for small N, almost sorted data	Yes
Mergesort	Θ(N)	Θ(N log N)*	Fastest stable sort	Yes
Random Quicksort	Θ(log N)	Θ(N log N)* expected	Fastest compare sort	No
Counting Sort	Θ(N+R)	Θ(N+R)	Alphabet keys only	Yes
LSD Sort	Θ(N+R)	Θ(WN+WR)	Strings of alphabetical keys only	Yes
MSD Sort	Θ(N+WR)	Θ(N+R) (best) Θ(WN+WR) (worst)	Bad caching (61C)	Yes

Previous
35.2 LSD Radix Sort

Next
35.4 Summary

Last updated 1 year ago