Merge Sort Performance

2 min

Merge sort was unique for its time in that the best, worst, and average time complexity are all the same: Θ(N*log(N)). This means an almost-sorted list will take the same amount of time as a completely out-of-order list. This is acceptable because the worst-case scenario, where a sort could stand to take the most time, is as fast as a sorting

Preview: Docs Loading link description

algorithm

can be.

Some sorts attempt to improve upon the merge sort by first inspecting the input and looking for "runs" that are already pre-sorted. Timsort is one such algorithm that attempts to use pre-sorted data in a list to the sorting algorithm's advantage. If the data is already sorted, Timsort runs in Θ(N) time.

Merge sort also requires space. Each separation requires a temporary

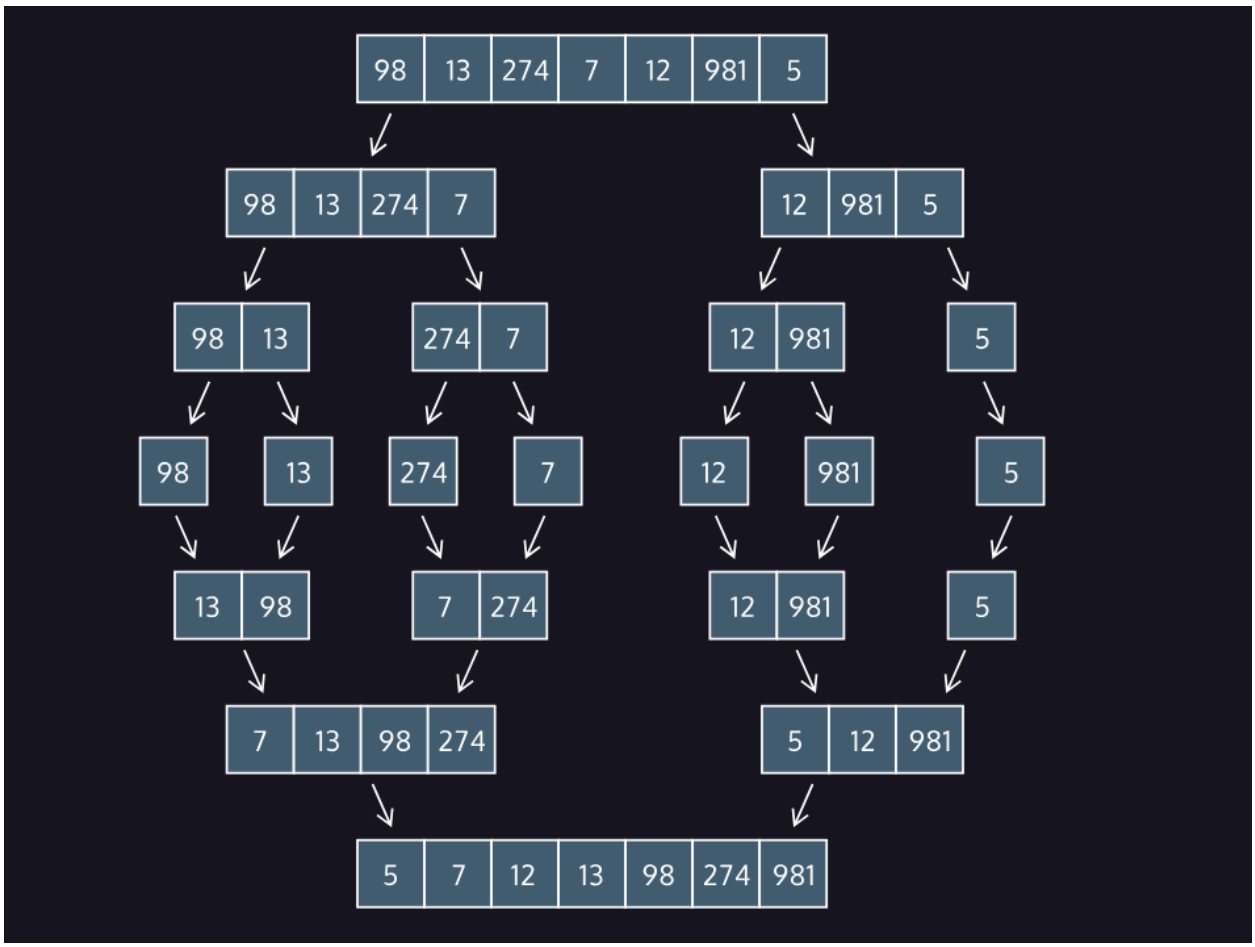
Preview: Docs Loading link description

array

, and so a merge sort would require enough space to save the whole of the input a second time. This means the worst-case space complexity of merge sort is O(N).

Instructions

Why does merge sort require so much space? Would it be possible to write an efficient sort that doesn't require any additional space? Can you think of any trade-offs that would need to be made?


98 13 274 7 12 981 5
98 13 274 7
12 981 5
98 13
274 7
12 981
5
98
13
274
7
12
981
5
13 98
7 274
12 981
5
7 13 98 274
5 12 981
5 7 12 13 98 274 981