Courtney Kelly

Mergesort

The merge sort algorithm creates a balanced binary tree referred to as the merge sort tree. It's height is log(n). At each level the algorithm does n work (the length of the vector at that level). This process occurs even in the worst case scenario, so even in the best, average, and worst case the run times are still O(nlogn)

1. Worst Case: O(nlogn)
2. Average Case: O(nlogn)
3. Best Case: O(nlogn)
4. The algorithm is stable because it preserves the order of equal elements. It does a level order traversal of the tree, so it processes the left child before the right and it doesn't reorder the equal elements within their levels.

Quicksort

1. Worst Case: O(n^2) The worst case occurs when the pivot is the unique minimum or maximum in the list. If this occurs at each iteration, the binary tree will only have right children or only have left children. So at depth 0, it does n work, at depth 1 it does n-1 work, at depth 2 it does n-2 work, … all the way to a depth of n-1 where it does 1 comparison.
2. Average Case: O(nlogn) On average, the algorithm creates a binary tree. If the pivots aren't poorly chosen, then the tree will be relatively balanced. Similarly to merge sort, the height of the tree will be log(n), and it does n work at each level, giving an average run time of O(nlogn)
3. Best Case: In most best cases, the run time is still O(n log n). However, if the pivot is chosen through partitioning like in the deterministic quick select method, then Quicksort has the potential best case run time of O(n) !
4. The algorithm is stable because it preserves the order of equal elements. It does a level order traversal of the tree, so it processes the left child before the right and it doesn't reorder the equal elements within their levels.