Adam Ryder

CS4050

Dr. Gurka

Questions

Bubble sort

* Bubble sort will be noticeably slower the larger n gets. It is a true O(n^2) algorithm, unless the data is already sorted in which case, early stop bubble sort will only be O(n). Alternating shaker sort is only marginally better than regular bubble sort, but still O(n^2).
* Best case is O(n) on an already sorted array for stop early and worst case and average case of O(n^2)

Insertion Sort

* Insertion sort is also O(n^2) but is usually faster than bubble sort because it can use shifts instead of swaps.
* Average and worst case is O(n^2) and best case O(n) for already sorted.
* The shift improvement makes this much better than swapping, but still average case O(n^2)
* The binary search method makes the search and insert routine nlog(n) which will make the best case of this sort nlogn, and average/worst case O(n^2).

Selection Sort

* Selection sort is very easy to understand but it is not very efficient. It is a true n^2 algorithm where best case, worst case and average case are all n^2 no matter what the data set is like.
* A small tweak would be to loop through bi-directionally, but that only eliminates the loop overhead and does not reduce swaps or comparisons.

Merge sort

* Merge sort is nLogn best,worst, and average case. But it is not in place and will require O(n) space to complete because you have to copy arrays.

Quicksort

* Quicksort is best and average case of O(nlogn) and worst case of (n^2) when the elements are sorted already.
* It is generally quicker than merge sort.
* Some improvements are to add insertion sort or selection sort at the Kth element. This decreases the recursive calls for smaller sets of size K and can be more efficient because insertion and selection are better for very small data sets. It also helps quicksort deal with somewhat sorted data. The big O doesn’t change for this improvement.
* Another improvement is when picking a pivot, try and choose a value that is going to be in the middle of the set. You can do this my getting the median of the first, middle and last element. This will help when data sets are more spread out because your pivot will generally be in the middle of the partition. The big O doesn’t change for this improvement.