Recurrence Relation for Sorting Algorithms

Recurrence Relation

$$T(N) = T(N-1) + N - 1$$
$$T(N) = T(N-1) + N \ (ignore\ constant)$$

N is the number of steps required to place the largest element at the end of the array and $T(N-1)$ signifies that time required to sort the rest of $N-1$ elements.

Recurrence Relation for Selection Sort

In each iteration of selection sort, we search for the smallest element and place it at the beginning of the array. After that, we increment the beginning position of the array and again search for the smallest element among the rest of the elements and repeat.

Searching the smallest element in the array takes $N-1$ comparisons.

Recurrence Relation

$$T(N) = T(N-1) + N - 1$$
$$T(N) = T(N-1) + N$$

N is the time required to search the smallest element in the array and place it at the beginning of the array. $T(N-1)$ is the time required to sort the rest of the $N-1$ elements.

Recurrence Relation for Quick Sort

Pseudo Code for Quick Sort

```
function quick_sort(array, start, end)
    if(start<end)
        pivot = partition(array,start,end)
        quick_sort(array,start,mid-1)
        quick_sort(array,mid+1,start)
    end if
end function
```

Quicksort first chooses a random element **X** as the pivot and partition the array in 2 parts such that the elements smaller than **X** are on the left side and elements greater than **X** are on the right side. After partition, **X** is in the correct position. Then we recursively sort the elements on the left side **X** and elements on the right side of **X**.

The partition requires N comparisons.

Recurrence Relation

Best Case

In the best-case scenario, the pivot will split the array into 2 equal parts. Therefore, the recurrence relation will be

$$T(N) = 2 * T(N/2) + N$$

N comparisons to find the pivot position and $2 * T(N/2)$ to sort the two partitions of the array.

Worst Case

In the worst case, the pivot will split the array of size N such that one part will contain no element, and the other party will contain $N - 1$ elements. This happens if all the elements are either smaller or greater than the pivot element.

$$T(N) = T(N - 1) + N$$