Okan Torun
1801042662

Q-1) To cut an n meter steel into 1 meter pieces and apply minimal cutting, we must first cut the steel in the middle and than repeat the cut in the middle by placing the resulting pieces one under the other.

Anlyze

$T_B(n) = \Theta(1)$ → if the length of the steed is less than 2meters

$T_W(n) = T(n/2) + 1$

Master Thorem

$a = 1$
$b = 2$
$f(n) = 1$

$\Rightarrow n^{\log_2 1} = 1$ so that $n^{\log_b a} \in \Theta(f(n))$

$T(n) = \Theta(n^{\log_b a} . \log n) \Rightarrow \underline{T(n) = \Theta(\log n)}$

Q-2) To find the worst and best experimental results, I sorted the results from smallest to largest using the merge sort algorithm and printed the best and worst results.

Okan Toran
1801042662

Q-2-cont.)

Analyze; The best case, worst case and avrage case cases of merge sort are $O(n.\log n)$.

$T(n) = 2T(\frac{n}{2}) + n$

Solution with master Theorem

$a=2$
$b=2$
$f(n)=n$

$n^{\log_b a} = f(n) \Rightarrow n^{\log_2 2} = n \Rightarrow n^{\log_b a} \in \Theta(f(n))$

$\boxed{T(n) = O(n.\log n)}$ for best and worst case

Q-3) I use Quick select algorithm. The algorithm is similar to QuickSort. Difference is, instead of recurring for both sides (after finding pivot) it recurs only for the part that contains the k-th smallest element. if index of partitioned element is more than k, the we recur for left part.

Analyze;

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

Solution with backward

$$\left.\begin{aligned} T(n) &= T(n-1) + n \\ T(n-1) &= T(n-2) + n-1 \\ &\vdots \\ T(2) &= T(1) + 2 \\ T(1) &= 0 \end{aligned}\right\} \quad T(n) = n + (n-1) + (n-2) \ldots + 4 + 3 + 2$$

$$= \frac{n.(n+1)}{2} \Rightarrow \boxed{T(n) = O(n^2)}$$

Okan Torun
1801042662

Q-4) The array is divided into two equal parts until the best case is reached

The FindPairs function was written to find reverse-ordered pairs on the right and left sides of the array

I created two indexes after the right and left of the index were combined. if a[i] greater than a[j] then there are reverse order.

Analyze:

$T(n) = 2T\left(\frac{n}{2}\right) + n$

Solution with master Theorem;

$a=2$
$b=2$
$f(n)=n$

$\Rightarrow \quad n^{\log_b a} = f(n) \Rightarrow n^{\log_2 2} = n \Rightarrow n^{\log_b a} \in \Theta(f(n))$

$T(n) = O(n^{\log_b a} . \log n) \Rightarrow T(n) = O(n . \log n)$

Okan Toru
18010426[illegible]

Q-5) Brute force: In the brute force algorithm the base number is multiplied by itself in a cycle equal to the power of the number and the exponential operation is found.

Analyze

$$\sum_{i=1}^{n} 1 \Rightarrow T(n) = \Theta(n)$$

loop

Divide and conquer

In divide and conquer algorithm, subproblems are obtained by dividing the power by two each time. The recursive continues until the power is 0.

Analyze

$$T(n) = T(n/2) + 1$$

Solution with master theorem,

$a=1$ $\quad n^{\log_b a} = f(n) \Rightarrow n^{\log_2 1} = 1 \Rightarrow n^{\log_b a} \in \Theta(f(n))$

$b=2$

$f(n)=1$ $\quad T(n) = O(n^{\log_b a} \cdot \log n) \Rightarrow T(n) = O(\log n)$