Quick Sort.

Algorithm quicksort (P, q)
{
    if (P < q) then
    {
        j = Partition (a, P, q+1)
        quicksort (P, j-1)
        quicksort (j+1, q)
    }
}.

DATE

Algorithm Partition (a, m, p)
// a is global array 1st element in array is a[m] and last element is a[p-1]. Initially Pivot is a first element, t = a[m]. After completion of algorithm it returns position j for some j between m and p-1.
a[k] ≤ t for m ≤ k < q, a[k] ≥ t for q < k < p. q is returned. Set a[p] = ∞

```
{ v = a[m] ; i = m ; j = p ;
{

   do
    {

      do
       { i = i+1;
       } while (a[i] ≤ v) && (i < p)

      do
       { j = j-1;
       } while (a[j] ≥ v) && (j > m)

     if (i < j) then swap (a, i, j);
   } while (i ≤ j)

  a[m] = a[j] ; a[j] = v ;
  return j ;
}
```

Worst case time Complexity of Quick sort

$$T(n) = \begin{cases} a & \text{for } n = 1 \\ T(n-1) + cn & \text{otherwise } n > 1 \end{cases}$$

$$T(n) - T(n-1) = cn \quad \text{—} ①$$
$$2T(n-1) - 2T(n-2) = 2c(n-1) \quad \text{—} ②$$
$$T(n-2) - T(n-3) = c(n-2) \quad \text{—} ③$$

① − ② + 3

$$T(n) - 3T(n-1) + 3T(n-2) - T(n-3) = 0$$

Characteristic Polynomial

$$x^3 - 3x^2 + 3x - 1 = (x-1)^3$$

$\therefore r_1 = 1, \ r_2 = 1, \ r_3 = 1$

General solution is

$$T(n) = C_1 \cdot 1^n + C_2 \cdot n \cdot 1^n + C_3 \cdot n^2 \cdot 1^n$$
$$T(n) = C_1 + nC_2 + n^2 C_3$$

$$T(1) = a = C_1 + C_2 + C_3 \quad \text{—} ①$$
$$T(2) = a + 2C = C_1 + 2C_2 + 4C_3 \quad \text{—} ②$$
$$T(3) = a + 2C + 3C = C_1 + 3C_2 + 9C_3 \quad \text{—} ③$$

$\therefore \ C_2 + 3C_3 = 2C$ ② − ①

$C_2 + 5C_3 = 3C$

$\therefore \ 2C_3 = C$

$C_3 = C/2$

$C_2 = C/2$

$C_1 = (a - C)$

$$\therefore T(n) = (a-c) + n \cdot \frac{c}{2} + n^2 \cdot \frac{c}{2}$$

$$\therefore T(n) = O(n^2).$$

Average case time Complexity of Quick sort.

Sahni

$$C_A(n) = n+1 + \frac{1}{n} \sum_{1 \le k \le n} C(k-1) + C(n-k) \quad \text{--- (1)}$$

The number of comparisons required by Partition on its first call is $n+1$.

Multiply both side of eqn (1) by n.

$$nC(n) = n(n+1) + 2[C(0) + C(1) \cdots C(n-1)] \quad \text{--- (2)}$$

Replace n by $n-1$ in (2).

$$(n-1)C(n-1) = (n-1)n + 2[C(0) + C(1) \cdots C(n-2)] \quad \text{--- (3)}$$

Substracting (3) from (2), we get.

$$nC(n) - (n-1)C(n-1) = 2n + 2C(n-1)$$

$$= nC(n) - nC(n-1) + C(n-1) = 2n + 2C(n-1)$$

$$= nC(n) - nC(n-1) = 2n + C(n-1)$$

$$= nC(n) = 2n + (n+1)C(n-1) \quad \text{--- (4)}$$

Devide above equation (4) by $n(n+1)$

$$= \frac{C(n)}{n+1} = \frac{2}{n+1} + \frac{C(n-1)}{n}$$

Repeatedly using this equation to substitute for $C(n-1)$, $C(n-2) \cdots$ we get

DATE

$$\frac{c(n)}{n+1} = \frac{c(n-2)}{n-1} + \frac{2}{n} + \frac{2}{n+1} \quad \text{--- (5)}$$

$$= \frac{c(n-3)}{n-2} + \frac{2}{n-1} + \frac{2}{n} + \frac{2}{n+1}$$

$$\vdots$$

$$= \frac{c(1)}{2} + 2 \sum_{3 \le k \le n+1} (1/k)$$

$$= 2 \sum_{3 \le k \le n} (1/k) \leftarrow (6)$$

Since $\sum_{3 \le k \le n+1} (1/k) \le \int_2^{n+1} \frac{1}{x}\, dx = \log_e (n+1) - \log_e 2$

$\therefore$ (6) will become.

$$c(n) \le 2(n+1)\left(\log(n+1) - \log 2\right)$$

$$= O(n \log n)$$