**CS502 Assignment NO.1 2020**

**Question no 01: (10: Marks)**

Apply quick sort algorithm on below given array by taking value (30) as pivot element and write down the resulted arrays till first pass only.

## Solution

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | | | | | | | | |
| 30 | 48 | 10 | 20 | 70 | 15 | 80 | 40 | ∞ |
| l=low  i |  |  |  |  |  |  |  | h=high  j |

Increment i until you find the element greater than pivot and decrement j until you find the element smaller than the pivot, we get

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | | | | | | | | |
| 30 | 48 | 10 | 20 | 70 | 15 | 80 | 40 | ∞ |
| L | i |  |  |  | J |  |  | h |
|  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | | | | | | | | |
| 30 | 15 | 10 | 20 | 70 | 48 | 80 | 40 | ∞ |
| l | i |  |  |  | j |  |  | h |

Swap i and j, we get

Increment i until you find the element greater than pivot and decrement j until you find the element smaller than the pivot, we get

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | | | | | | | | |
| 30 | 15 | 10 | 20 | 70 | 48 | 80 | 40 | ∞ |
| l |  |  | j | i |  |  |  | h |

Now swap j with pivot and we get on its sorted position

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | | | | | | | | |
| 20 | 15 | 10 | 30 | 70 | 48 | 80 | 40 | ∞ |
| l |  |  |  |  |  |  |  | h |

Now the Partition Algorithm is as below:

Partition (l, h)

{

do

{

Pivot = A[l]; i = l, j = h;

while (i < j)

{

i++;

} while (A[i] <= pivot);

do

{

j--;

} while (A[j] > pivot);

if (i < j)

{

Swap (A[i], A[j]);

}

}

Swap (A[l], A[j]); return j;

}

We get the following resulted array after the first pass (means partition algorithm)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | | | | | | | | |
| 20 | 15 | 10 | 30 | 70 | 48 | 80 | 40 | ∞ |
| l |  |  |  |  |  |  |  | h |

Now apply following quick sort algorithm on resulted array by taking value (30) as pivot element

QuickSort (l, h)

{

if (l < h)

{

j = partition (l, h); QuickSort (l, j); QuickSort (j+1, h);

}

}

# Question no 02: (10: Marks)

Write down the recurrence relation of quick sort in case of average or normal case means when the pivot’s element location is the middle of array.

## Solution

The recurrence relation of quick sort in case of average or normal case is as given below:

T(n) = 1 if n = 1, 2T(n/2) + n otherwise

T(n) = 2T(n/2) + O(n)

= 2(2T(n/4) + O(n)

= 4T(n/4) + O(n)

= 4(2T(n/8) + O(n)

= 8T(n/8) + O(n)

= 8(2T(n/16) + O(n)

= 16T(n/16) + O(n)

If n is a power of 2 then let n = 2k or k = log n. T(n) = 2kT(n/(2k)) + O(n)

= 2kT(n/(2k)) + O(n)

= 2(log n)T(n/(2(log n))) + n (log n)

= 2(log n)T(n/n) + n (log n)

= nT(1) + n log n

= n + n log n

Or T(n) = O (n log n)