

Lecture Notes

on

Quick Sort



(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

July 2020

(Be safe and stay at home)

Sorting

- sorting is an algorithm that arranges the element of a list in a certain order (either ascending or descending)
- sometimes, sorting significantly reduces the problem complexity.
- These algorithms are very much used in computer algorithms, database algorithms and many more.

Classification of sorting algorithms

it can be done via:

- 1 By number of comparisons
- 2 By number of swaps
- 3 By memory usage
- 4 By recursion
- 5 By stability

other classification

- internal sort
- external sort

Quick Sort

- its a based on divide and conquer approach.
- In quick-sort, the division into two sub-arrays is made so that the sorted sub-array do not need to be merged later.
- The sorting is done by picking some element of an input array (pivot element) and then reorder other elements so that all the elements appearing before **pivot** in $a[1 : n]$ are less than or equal to pivot
- & all elements appearing after pivot are greater than or equal to pivot.
- This process is referred as **partitioning**.

```
Quick-sort(a[],p,q)
{
    if(p==q)
        return(a);
    else
    {
        pivot=partition(a,p,q);
        Quick-sort(a,p,pivot-1);
        Quick-sort(a,pivot+1,q);
    }
    return(a);
}
```

```

partition(a,m,p)
{
    v=a[m];
    i=m;
    j=p;
    repeat
    {
        Repeat
            i=i+1;
            until(a[i]>=v);
        Repeat
            j=j-1;
            until(a[j]<=v);
    }
    if(i<j)then
        interchange(a[i],a[j]);
    }
    until(i>=j)
    interchange(a[j],a[m]);
    return j;
}

```

Running example

65 70 75 80 85 60 55 50 45

65 70_i 75 80 85 60 55 50 45_j

65 45 75_i 80 85 60 55 50_j 70

65 45 50 80_i 85 60 55_j 75 70

65 45 50 55 85_i 60_j 80 75 70

65 45 50 55 60_j 85_i 80 75 70

60 45 50 55 **65** 85 80 75 70

Analysis

Best-case:

each partition splits array in halves & gives recurrence relation as:

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

we can easily apply master's method here.

$$T(n) = O(n \log_2 n)$$

Worst-case:

each partition gives unbalanced splits & we get recurrence relation as:

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

after solving it by *substitution method*

$$T(n) = O(n^2)$$

avg-case:

we don't know where the split happens & recurrence relation for avg. case is not defined. it is depend on partition.

$$T(n) = O(n \log_2 n)$$

practice questions

What would be the output after calling quick-sort first time?

1. 54 26 93 17 77 31 44 55 20

2. 40 70 20 30 10 60 50

Q & A?

Queries are welcome on slack channel
for discussion