

QUICKSORT

Quicksort is an algorithm of the divide and Conquer type, i.e. the problem of sorting a set is reduced to the problem of sorting two smaller sets.

Ex Suppose A is the list of 12 numbers.

(44) 33 11 55 77 90 40 60 99 22 88 66 ←

→ At first we choose the first number 44 as pivot. Beginning with the last number 66, scan the list from right to left. Compare each no with 44 and stop at the first number which is less than 44. Interchange it.

(22) 33 11 55 77 90 40 60 99 (44) 88 66 →

→ Now beginning with 22, next scan the list in the opposite direction i.e. from left to right, comparing each element with 44 and stop at the first no which is greater than 44. Interchange this.

22 33 11 (44) 77 90 40 60 99 (55) 88 66 ←

→ Begin with 55, scan right to left until meeting the first no less than 44. interchange it.

22 33 11 (40) 77 90 (44) 60 99 55 88 66

→ Now start with 40, scan left to right.

22 33 11 40 (44) 90 (77) 60 99 55 88 66
First sublist Second sublist

⇒ Now 44 reached to its real position.
Then solving these two sublists individually by using the above method we get the sorted array.

Pseudocode

```
void quicksort (int a[], int p, int r)
{
    int q;
    if (p < r)
    {
        q = partition (a, p, r);
        quicksort (a, p, q);
        quicksort (a, q+1, r);
    }
}

int partition (int a[], int p, int r)
{
    int x, i, j, temp;
    x = a[p];
    i = p-1;
    j = r+1;
    while (1)
    {
        do
        {
            j = j-1;
        } while (a[j] > x);

        do
        {
            i = i+1;
        } while (a[i] < x);

        if (i < j)
        {
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        }
        else
            return (j);
    }
}
```