Time Complexity

Best case

$$\frac{n}{2} = 1$$

$$\frac{n}{2^k} = 1$$

$$n = 2^k$$

$$k = \log_2 n$$

$$O(n \log n)$$

Worst case

2, 4, 8, 10, 16, 18, 17

n ——— 1, 7

n-1 ——— 2, 7

n-2 ——— 3, 7

n-3 ——— 4, 7

$$\frac{n(n+1)}{2}$$

$$= O(n^2)$$

```
algo Quicksort (a[], lb, ub)
{
  if (lb<ub)
   {
     loc = Partition (a, lb, ub)
     Quicksort (a, lb, loc-1)
     Quicksort (a, loc+1, ub)
   }
}

algo Partition (a[], lb, ub) return integer
 {
   Pivot = a[lb]
   start = lb,
   end = ub,
   while (start <end)
    {
     while (a [start] <= Pivot && start <end)
        {
          start = start +1
        }
     while (a[end] > Pivot)
        {
          end = end-1
        }
     if (start <end)
      {
      swap (a[start], a[end])
      }
   }
   a[lb] = a[end]
   a[end] = Pivot   return end
}
```

Ex:-

1	2	3	4	5	6	7	8
60	40	20	95	90	30	50	70
P	S						E

Step 1

```
while (1<8)
{
while (60<=60 && 1<8)
  {
     start=start+1
       (40<=60 && 2<8)
     start=start+1
       (20<=60 && 3<8)
       start=start+1
       (95<=60 && 4<8)
  }
```

Pivot	start	end
60	~~1~~ ~~2~~ ~~3~~ 4	8

```
while (a[end] > Pivot)
        70>60
      end=end-1
      50>60 False.
```

Pivot	start	end
60	4	~~8~~ 7

```
if (4<7)
  swap (a[start], a[end])
```

1	2	3	4	5	6	7	8
60	40	20	50	90	30	95	70

step2 (start<end)

(4<7)

Piv	start	end
60	4 / 5	7 / 6

(50<=60 && 4<7)

start = 4+1 = 5

(90<=60 && 5<7) False.

while(95>60)

(30>60) False

if (start<end)

(5<6)

swap(a[start],a[end])

1	2	3	4	5	6	7	8
60	40	20	50	30	90	95	70

step3 (start<end)

(5<6)

Piv	start	end
60	5 / 6	6 / 5

while(30<=60 && 5<6)

5+1

(90<=60 && 6<6) False

while (a[end]>Piv)

(90>60) True

end=end-1

(30>60) False

if (6 < 5) False

Piv	start	end (4)
60	6	5

a[lb] = a[end]

a[end] = Pivot

1	2	3	4	5	6	7	8
30	40	20	50	60	90	95	70

Partition-1 | Pivot | Partition 2

1	2	3	4
30	40	20	50

while (1 < 4)

{

(30 <= 30 && 1 < 4)

Piv	start	end
30	1 → 2	4

(40 <= 30 && 2 < 4) False

whie (50 > 30)

(20 > 30) False

Piv	s	e
30		4 → 3

if (2 < 3)

{

swap (a[start], a[end])

1	2	3	4
30	20	40	50

while (2 < 3)

{

Piv	S	E
30	2 → 3	3 → 2

while (20 <= 30 && 2 < 3)

while (40 <= 30 && 3 < 3) False

while (40 > 30)

(20 > 30) false

if (3<2)

a[lb] = a[end]

a[end] = pivot.

20	30	40	50

Partition-1 sorted.

Partition-2

	1	2	3
	90	95	70

P	S	E
90	~~1~~ 2	3

while (1<3)

{

while (90<=90 && 1<3)

while (95<=90 && 2<3) False

while (70>90) False

if (2<3)

{

swap (a[start], a[end])

(95, 70)

	1	2	3
	90	70	95

P	S	E
90	~~2~~ 3	~~3~~ 2

while (2<3)

{

while (70<=90 && 2<3)

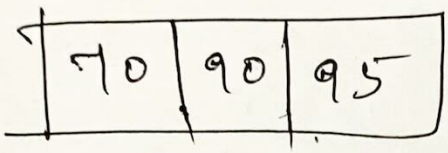
while (95<=90 && 3<3) False

while (95>90)

if (3 < 2) False

a[lb] = a[end]

a[end] = Pivot.



Partition-2 sorted order.