Quicksort Time Complexity

```
Quicksort(A, P, r)
  if P > r:
    q = Partition(A, P, r)
    Quicksort(A, P, q-1)
    Quicksort(A, q+1, r)
Partition(A, P, r)
  x = A[r]
  i = P-1
  for j = P to r-1
    if A[j] ≤ x
      i = i+1
      Swap A[i] with A[j]
  swap A[i+1] with A[r]
  return i+1
```

$$T(n) = T(\text{Partition}) + T(n - \text{Partition}) + T(\text{finding Pivot})$$

P : Pivot

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

$$= \frac{1}{n}\sum_{i=1}^{n-1} T(P) + \frac{1}{n}\sum_{i=1}^{n-1} T(n-P) \qquad T(P) \approx T(n-P)$$

$$n \cdot T(n) = 2\sum_{i=1}^{n-1} T(P) \qquad n = n-1 \qquad (n-1)T(n-1) = 2\sum_{i=1}^{n-1} T(P)$$

$$n \cdot T(n) - (n-1)T(n-1) = 2T(n-1) + Cn^2 + C(n-1)^2$$

Quicksort Time Complexity

$nT(n) = T(n-1)(2+n-1) + 2Cn - C$ ← Constant

$\frac{nT(n)}{n(n+1)} = \frac{T(n-1)(n+1)}{n(n+1)} + \frac{2Cn}{n(n+1)}$

$n = n-1$

$T(n-1)/n = T(n-2)/n-1 + 2C/n \qquad n = n-2$

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

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

$\frac{T(n)}{n+1} = \frac{T(1)}{2} + 2C\left(\frac{1}{n-1} + \frac{1}{n} + \frac{1}{n+1} + \cdots\right)$

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

$T(n) = 2C\log n \cdot (n+1)$

$T(n) = \log n(n+1)$

$= O(n\log n)$

Average and best case are the same

$T(n) = \Theta(n\log n)$