QUICKSORT ANALYSIS

Worst case timing analysis : $T(n) = O(n^2)$

Quicksort gives a worst case behaviour when the set is **already sorted.**

Assume the set of elements to be sorted is already sorted and in ascending order.

Consider the first call on partition:

The left_to_right pointer stops at the second element with the cost of one comparison.

The right_to_left pointer stops at the first element after n comparisons.

The position of the pivot element remains at 1, at the cost of n +1 comparisons.

Now the set to be sorted contains a left part which is null set and the right part which has n –1 element.

Hence $T(n) = n+1 + T(n-1)$ with $T(0)=T(1)=1$

Solving this recurrence by the substitution method:

$T(n)$	=	$T(n-1)$	+	$(n+1)$
$T(n-1)$	=	$T(n-2)$	+	n
$T(n-2)$	=	$T(n-3)$	+	$(n-1)$
$T(n-3)$	=	$T(n-4)$	+	$(n-2)$

……………………………………………..

…………………………………………

$T(3) \quad = \quad T(2) \quad + \quad 4$

$T(2) \quad = \quad T(1) \quad + \quad 3$

$T(1) \quad = \quad T(0) \quad + \quad 2$

Summing all the left hand sides of the above equations:

$T(n) \; + \; [T(n-1) + T(n-2) + \text{-------------------} + T(2) + T(1)]$

Summing all the right hand sides of the above equations:

$[T(n-1) + T(n-2) + \text{-------------------} + T(2) + T(1)] + [(n+1) + (n) + (n-1) + \text{----} + 3 + 2] +$

$T(0)$

Equating the two sides

$T(n) = [(n+1) + (n) + (n-1) + \text{----} + 3 + 2] + T(0)$

$= 1 + 2 + 3 + 4 + \text{----------} + n + (n+1)$

$= (n+1)(n+2)/2$

$= (n^2 + 2n + 2)/2$

$< n^2/2$

$= O((n^2)$

WORST CASE

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

Average case analysis of quicksort: $T(n) = O(n \log n)$

Position of pivot	Left of pivot	Right of pivot	$T(n_{left})$	$T(n_{right})$
1	0	n-1	T(0)	T(n-1)
2	1	n-2	T(1)	T(n-2)
3	2	n-3	T(2)	T(n-3)

-----				
n-2	n-3	2	T(n-3)	T(2)
n-1	n-2	1	T(n-1)	T(1)
n	n-1	0	T(n-1)	T(0)

The number of comparisons for first call on partition:

Assume left_to_right moves over k smaller element and thus k comparisons.

So when right_to_left crosses left_to_right it has made n-k+1 comparisons.

So first call on partition makes n+1 comparisons.

The average case complexity of quicksort is

T(n) = comparisons for first call on quicksort

+

$\{\Sigma_{1<=nleft,nright<=n} [T(n_{left}) + T(n_{right})]\}n$

$= (n+1) + 2\ [T(0) + T(1) + T(2) + \text{-----} + T(n-1)]/n$

$nT(n) = n(n+1) + 2\ [T(0) + T(1) + T(2) + \text{-----} + T(n-2) + T(n-1)]$

$(n-1)T(n-1) = (n-1)n + 2\ [T(0) + T(1) + T(2) + \text{-----} + T(n-2)]$

Subtracting both sides:

$nT(n) - (n-1)T(n-1) = [\ n(n+1) - (n-1)n] + 2T(n-1)$

$= 2n + 2T(n-1)$

$nT(n) = 2n + (n-1)T(n-1) + 2T(n-1)$

$= 2n + (n+1)T(n-1)$

$T(n) = 2 + (n+1)T(n-1)/n$

The recurrence relation obtained is:

$T(n)/(n+1) = 2/(n+1) + T(n-1)/n$

Usng the method of subsititution:

$T(n)/(n+1) = 2/(n+1) + T(n-1)/n$

$T(n-1)/n = 2/n + T(n-2)/(n-1)$

$T(n-2)/(n-1) = 2/(n-1) + T(n-3)/(n-2)$

$T(n-3)/(n-2) = 2/(n-2) + T(n-4)/(n-3)$

$T(3)/4 = 2/4 + T(2)/3$

$T(2)/3 = 2/3 + T(1)/2$

$T(1)/2 \quad = \quad 2/2 \quad + \quad T(0)$

Adding both sides:

$T(n)/(n+1) + [T(n-1)/n + T(n-2)/(n-1) + \text{-------------} + T(2)/3 + T(1)/2]$

$= [T(n-1)/n + T(n-2)/(n-1) + \text{-------------} + T(2)/3 + T(1)/2] + T(0) \quad +$

$[2/(n+1) + 2/n + 2/(n-1) + \text{----------} + 2/4 + 2/3]$

Cancelling the common terms:

$T(n)/(n+1) = 2[1/2 + 1/3 + 1/4 + \text{--------------} + 1/n + 1/(n+1)]$

$= 2\Sigma_{2<k<=(n+1)} (1/k)$

$[= 2H_k, H_k$ is called the kth harmonic number$]$

$<2 \int dk/k$ for k varying from 2 to $(n+1)$

$<2[\log_e(n+1) - \log_e 2]$

Hence $T(n) = O([(n + 1) 2[\log_e(n+1) - \log_e 2])$

$= O(n \log n)$

AVERAGE CASE
T(n) = O(n log n)