ADS Homework 5

Aashish Paudel

March 14, 2019

Problem 1

(a, b, c)

The code for the problems is attatched with zip file. Filename: quick_sort.py

(d)

The code for generation of values is attached with zip file. Filename: data_generation.py

The average time for different algorithms is as follows:
Lomuto partition: 1.576164960861206e-05
Hoare partition: 1.408672332763672e-05
Lomuto with best pivot: 1.4030117988586426e-05
Hoare with best pivot: 1.3973305225372314e-05

We can see that the time decreases with each variation. This is because hoare uses less swaps than lomuto at total. On the other hand, if we choose our pivot wisely by considering three elements (beginning, end, and the middle) and select the best fitting pivot amongst those three, we are comparitively less likely to run into the uneven splits that move our algorithm more towards best case i.e.**farther from $\theta(n^2)$ and closer to $\theta(nlg(n))$** Thus, the latter 2 are better than first two. Also, hoare performs better than lomuto with best pivot too, becaue of the less swaps it uses.

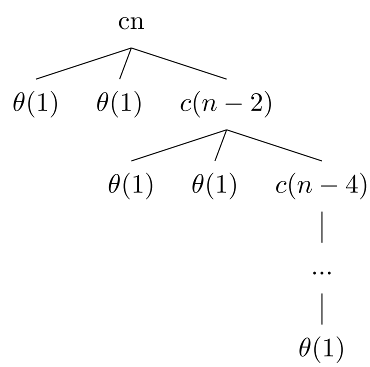
Problem 2

(a, c)

The code for the problems of values is attached with zip file. A and C are done in the same file. The implementation is described within .py file. The next subproblem (b) does not contain any inference from the solution of (c). Filename: q_variant.py

(b)

The general time complexity: $T(n) = 3T(n/3) + \theta(n)$
Time complexity of worst case:
Consider the array already sorted.
Then, the time complexity becomes: $\mathbf{T(n) = T(0) + T(1) + T(n-2) + cn}$



Here, the height of the tree is length of the series $n-2, n-4, n-6, ...4, 2$
Thus, $h = n/2$
$or, \theta(n) = \sum_{i=1}^{n} 2i = n(n+1) = \theta(n^2)$

$\because \theta(\mathbf{n}) = \mathbf{n^2}$

Time complexity of best case:
The best case is when the array is divide into equal $n/3$ sequences at every iteration.
Thus, the time complexity: $T(n) = 3T(n/3) + \theta(n)$

Applying master method:
$\theta(n) = n^{log_3 3} = n$
Thus, the time complexity: $\theta(\mathbf{n^{log_3 3} lgn}) = \mathbf{nlg(n)}$

Problem 3

We have,

$$lg(n!)$$
$$=lg(n.(n-1).(n-2).(n-3)...3.2.1) \qquad (i)$$

Upperbound:

$$(i) \leq lg(n^n) = nlg(n)$$
$$or, \mathbf{lg(n!) = O(nlg(n))}$$

Lower bound:

$$(i) \geq \lg(n^{\frac{n}{2}}) = \frac{1}{2}nlg(n)$$
$$or, \mathbf{lg(n!) = \Omega(nlg(n))}$$

Thus, it can be concluded that: $\mathbf{lg(n!)} = \theta(\mathbf{nlg(n)})$