CS344 Spring 2014 Quiz 1 Solutions

A problem called X can be solved with three different recursive algorithms A_1, A_2 and A_3. The recurrence for algorithm A_i is given by $T_i(N)$, where $T_i(N)$ is the cost (say, running time) of A_i on input of size N. Here are the recurrences:

- $T_1(N) = 2T_1(N/2) + 4N$.
- $T_2(N) = T_2(N-1) + 2N$.
- $T_3(N) = T_3(N-1) + \log N$.

Assume that $T_i(1) = 1$, for each i.

For each question, circle the correct answer (True or False). All comparisons are in the asymptotic sense. Explain your answers.

1. (**True**/False) A_1 is no worse than the other algorithms.
2. (**True**/False) A_2 is the worst algorithm.
3. (True/**False**) A_3 is strictly better than A_1.
4. (**True**/False) A_3 has the same cost as sorting using MergeSort.
5. (**True**/False) A_2 has the same cost as sorting using Insertion Sort.

Solving the Recurrences

Algorithm 1 For $T_1(N) = 2T_1(N/2) + 4N$, we use the Master Theorem for $T(N) = aT(N/b) + f(N)$ with $a = b = 2$ and $f(N) = 4N$. This is the case where $f(N) = \Theta(N^{\log_b a})$.
$\boxed{T_1(N) = \Theta(N \log N)}$

Algorithm 2 For $T_2(N) = T_2(N-1) + 2N$, we use the Substitution method with the *guess* $T_2(N) \leq N(N+1)$. Then, we *verify* this guess as follows:

$$T_2(N) = T_2(N-1) + 2N \leq (N-1)N + 2N = N^2 + N = N(N+1). \tag{1}$$

$\boxed{T_2(N) = O(N^2)}$

Algorithm 3 For $T_3(N) = T_3(N-1) + \log N$, we use the Substitution method with the *guess*: $T_3(N) \leq \log(N!)$. Then, we *verify* this guess as follows:

$$T_3(N) = T_3(N-1) + \log(N) \leq \log((N-1)!) + \log(N) = \log(N!), \tag{2}$$

since $\log(a+b) = \log a + \log b$ and $N! = N \times (N-1)!$.
$\boxed{T_3(N) = O(N \log N)}$