Australian National University
Department of Computer Science

# COMP3600/COMP6466 in 2009 – Assignment One

**Due:** 5pm Friday, September 18
**Late Penalty:** 20% per day

No programming is needed for this assignment. Put your work in the `COMP3600` box on the ground floor of the CSIT Building. The total marks for this assignment are 50 points for COMP3600 and 60 points for COMP6466.

**Question 1** (5 points).
Given the following sequence, order them into a sorted sequence in the order of growth when $n$ approaches infinity.

$$n^{3.5}, n \log\log n, n \log^3 n, n^{3/2} \log n, 2^{\ln n}, 3^{0.3n}, n^5, 2^{1/n}, 3^{100} \log n.$$

**Question 2** (12 points).
Let $f(n)$, $g(n)$ and $h(n)$ be three positive function. For each of the following statements, either prove that the statement is true or give an example for $f, g, h$ showing that the statement is false.

(a) $f(n) = O(g(n))$ and $g(n) = o(h(n))$ together imply that $f(n) = o(h(n))$;

(b) $\min\{f(n), g(n), h(n)\} = O(f(n) + g(n) + h(n))$;

(c) $f(n) = \Omega(g(n))$ and $g(n) = O(h(n))$ together imply that $f(n) = \Theta(h(n))$.

**Question 3** (8 points for COMP3600, 12 points for COMP6466).
Give an expression for each of the following sums using the $\Theta(\,)$ notation. In each case explain your reasoning clearly.

(a) $\sum_{k=1}^{n} k^{5/3}$.

(b) $\sum_{k=1}^{n} k^4/3^k$.

(c) (COMP6466 only) $\sum_{k=1}^{n} \lg^4 k/k$.

You might care to know that $\int (\log x)^4/x \, dx = C + \frac{1}{4}(\log x)^5$.

**Question 4** (12/50).
Give an asymptotic upper bound for $T(n)$ using the $O(\,)$ notation in each of the following recurrences. Justify your answers. Don't use *the Master Theorem.*

(a) $T(n) = 5T(n/3) + n^2$

(b) $T(n) = T(n/4) + n \log n$

(c) $T(n) = T(n/5) + T(2n/3) + n$

**Question 5** (13 points).
You are going on a long trip. You start on the road at mile post 0. Along the way there are $n$ hotels, at mile posts $a_1 < a_2 < \ldots a_n$, where each $a_i$ is measured from the starting point. The only places you are allowed to stop are at these hotels, but you can choose which of the hotels you stop at. You must stop at the final hotel (at distance $a_n$), which is your destination.

You would ideally like to travel 200 miles a day, but this may not be possible (depending on the spacing of the hotels). If you travel $x$ miles during a day, the *penalty* for that day is $(200 - x)^2$. You want to plan your trip so as to minimize the total penalty – that is, the sum, over all travel days, of the daily penalties.

Describe an efficient algorithm that determines the optimal sequence of hotels at which to stop, and analyse the running time of your algorithm. *Hint: use dynamic programming.*

**Question 6** (6 points for COMP6466 only).
A subsequence is *palindromic* if it is the same whether read from left to right or right to left. For instance, the sequence

$$A, C, G, T, G, T, C, A, A, A, A, T, C, G$$

has many palindromic subsequences, including $A, C, G, C, A$ and $A, A, A, A$ (on the other hand, the subsequence $A, C, T$ is not palindromic).

Devise an algorithm that takes a sequence $x[1 \ldots n]$ and return the (length of the) longest palindromic subsequence. It running time should be $O(n^2)$.