ELL781: Minor Test I

August 24, 2018

Maximum Marks: 20

1. Consider an undirected graph with 6 vertices and 14 edges, similar to the kind of graph we constructed to represent conflicting turns at a traffic intersection (there are no self-loops). Answer the following questions, with proof/justification:

   (a) How many vertices does the largest clique in the graph contain? **[1.5]**

   (b) Based on just the answer to part (a), what can you say about the minimum number of colours needed to colour this graph? Give a range of possible values for this. **[1]**

   (c) Can you apply some further reasoning (beyond just using the answer to part (a)) to give a definite value for the minimum number of colours needed? Explain/illustrate your reasoning clearly. If you can come up with a definite value, draw an example graph with the above properties and show a colouring using that many colours. **[3]**

2. We discussed two definitions of $\Omega(f(n))$ in class, which are repeated below to aid your memory:

   Definition I: $T(n)$ is $\Omega(f(n))$ if $\exists c > 0, n_0$ such that $\forall n \geq n_0, T(n) \geq cf(n)$.

   Definition II: $T(n)$ is $\Omega(f(n))$ if $\exists c > 0$ such that $T(n) \geq cf(n)$ for infinitely many values of n.

   (a) Consider

   $$T(n) = \begin{cases} n & \text{if } n \leq 100 \\ n^3 & \text{if } n > 100 \end{cases}$$

   Prove whether or not $T(n)$ is $\Omega(n^3)$, under each of Definition I and Definition II. **[2]**

   (b) Under Definition I, can it be shown in general that $T(n)$ is $\Omega(f(n))$ if and only if $f(n)$ is $O(T(n))$? If so, give a proof; if not, give a counter-example. **[4]**

   (c) Now answer the same question for Definition II. **[3]**

   [Hint: In proving an 'if an only if' claim, both directions need to be proved. First assume one side and show that it implies the other, and then do the converse.]

3. Consider the pointer-based implementation of the LIST abstract data type (*i.e.*, a linked list implementation). Write down:

   (a) A `struct` definition to represent the nodes/cells of the linked list. (Assume the list elements will be integers.) **[1.5]**

   (b) A corresponding implementation of the operation END(L), which takes an argument of the abstract data type LIST and returns the position following the last element in the list. (Assume the use of a header cell, and the corresponding notion of position, as discussed in class.) **[4]**

   [Exact C syntax is not necessary; C-like pseudocode is fine. However, the types of the different variables in your implementation should be made clear, including the input and return types of any functions.]