University of Waterloo
Waterloo, Ontario
CS 466/666
Midterm Examination – June 27, 2013
NOTE: The order of topics covered in 2014 differs from that of 2013 … so some of the questions here are on topics not yet covered

Time: 75 minutes **Instructor: J. I. Munro**

Surname:_________________________ **Initials:**____________ **ID.#:**_________________

Signature: _________________________

Do all questions in the space provided, using the back of the preceding page if necessary. This exam consists of 9 pages including this one. Your answers need to be justified only if it is explicitly stated so.

QUESTION	MAXIMUM	MARK
1	37	
2	9	
3	8	
5	21	
TOTAL	75	

1. [37 marks] Short answers

(a) [6 marks] Consider the problem of identifying graphs that **do not** contain a Hamiltonian cycle. Is this problem in NP or a related class? Briefly justify this claim.

(b) [8 marks] We also discussed a greedy algorithm to find a near optimal binary search tree. What are the **two** main advantages of this method over the dynamic programming approach even though the dynamic programming algorithm will probably give a better search tree?

(c) [8 marks] AVL trees (or other balanced search tree structures) are simpler to implement than splay trees and give a worst case guarantee of $\Theta(\lg n)$ behaviour, so what advantages do splay trees have in terms of behavior? Give **two** advantages.

(d) [5 marks] Splay trees involve performing three different types of rotations in moving an element to the root. It would seem simpler to just use the “zig” step (and never the “zig-zig” or “zig-zag”) to move the element all the way up. Why is this not done?

(e) [5 marks] It was shown that we can get a 2-approximation for the traveling salesman problem (with triangle inequality) by taking a tour that consists of following each edge of the minimum spanning tree in each direction (and then using the triangle inequality to convert this to a proper tour with no greater cost). What is the key relation between the minimum spanning tree and the optimal tour that permits us to claim a 2-approximation?

(f) [5 marks] Suppose we are given a Boolean expression in CNF in which each of n clauses has exactly 4 distinct literals (no literal is repeated in any one clause). We are interested in making an assignment of the variables, quickly, while trying to maximize the number of clauses that are true. What fraction of the clauses can you guarantee can be satisfied? Why?

2. [9 marks] Maxima

Suppose we are scanning a sequence of numbers looking for the maximum. Clearly this will take n-1 comparisons. Our interest, however, is in the number of times a new maximum is discovered. This could be as low as 1, if the maximum comes first, or as high as n, if the numbers are in increasing order.

a) [3 marks] If the elements are in random order (all permutations equally likely), what is the probability that the i^{th} element will be larger than all preceding it?

b) [6 marks] What is this expected number of "new maxima" in a randomly ordered sequence of length n? Give your answer up to an additive O(1) term and justify your claim.

3.[8 marks] Vertex Cover
Sketch a linear time algorithm for the special case that the graph is a tree.

5. [21 marks] Cluster graphs
A cluster graph is a collection of disjoint cliques. i.e. it is a graph in which each connected component contains all possible edges.
The purpose of this problem is to design fixed-parameter algorithms to determine whether an arbitrary n node graph can be made into a cluster graph by deleting at most k vertices. Here k is the parameter. We will also discuss the minimization version where the goal is to find the minimum number of vertices needed to delete to make the given graph a cluster graph.

a) [6 marks] Show that a graph G is a cluster graph if and only if there is **no** `induced path' on 3 vertices. I.e. there are no three vertices a, b and c in G such that there is an edge between a and b, and an edge between b and c, but there is no edge between a and c.

Cluster graphs Continued

b) [6 marks] Using the claim in the previous part, design an $O(f(k)n^c)$ algorithm to determine whether a given graph G can be made into a cluster by deleting at most k vertices. What is your f(k)?

Cluster graphs continued

c) [9 marks] Give an approximation algorithm to find the minimum number of vertices needed to delete to make the given graph cluster graph. What is the approximation ratio of your algorithm? Prove this.