Student ID: ____________ Student Name: ________________________

CompSci 369: Computational Science

Midterm Test 2010

1. Short answer questions:

   (a) Give an example of a very high level programming language and explain why Java is not one of them.

Python/Sage; Java only supports iterative/OO design at a logical step-by-step level.

   (***3 marks***)

   (b) How many different (structurally) Steiner/Parsimony trees are there with $n = 3$, 5 and 6 distinct leaves?

1, 3*5, 3*5*7

   (***3 marks***)

   (c) Name and informally describe three different string matching algorithm.

See lecture notes on string matching algorithms (6 given).

   (***3 marks***)

   (d) Explain the bisection method for numerically finding roots.

See lecture notes on numerical algorithms.

   (***3 marks***)

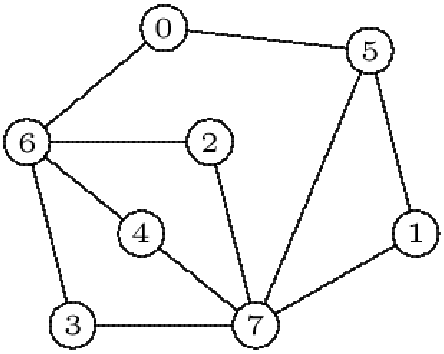
   (e) When is a memoized implementation of a dynamic program more suitable than an pure iterative implementation (and vise versa).

If few of the subproblems (say less than half) need to be evaluated then memoized algorithms are usually faster (assuming you have memory for the recursive calls)

   (***3 marks***)

2. Consider the following **Dominating Set Optimization Problem**.
Input: An undirected graph $G = (V, E)$.
Output: Minimum size of a dominating set $D \subseteq V$ such that each vertex $v \in V$ is either in the set D or has a neighbor in D.

(a) What is the minimum dominating set size and how many different sets of that size exist for the following graph.



The minimum dominating set is of size 2; there are *four* of them.
$\{0,7\}$, $\{1,6\}$ $\{5,6\}$ and $\{6,7\}$,

(***3 marks***)

(b) Give a simple backtracking algorithm that solves this problem.

See the backtracking template from the course lecture notes.

(***4 marks***)

(c) Help complete an Integer Programming formulation that solves this problem.
Let $x_i = 1$ iff vertex i is in the Dominating Set of $G = (V, E)$.

What is the optimization objective? (Hint: what formula to minimize.)

Minimize $\sum_{i=1}^{n} x_i$

(***2 marks***)

What are the optimization constraints? (Hint: one equation for each vertex.)

$x_i + \sum_{j \in N(i)} x_j \geq 1$

(***2 marks***)

(d) Give a Local Search algorithm that approximates this problem. (Explain the neighborhood and gradient descent aspects similar to the vertex cover local search example given in lectures.)

Start with all vertices in dominating set. Repeatedly remove a vertex such that at least one of its neighbors is still in the set.

(***4 marks***)

3. Computational Biology

(a) State the two assumptions of the Jukes Cantor Model.

i. All mutation rates are equal

ii. All residues are equally likely

(*1 marks*)

(b) Starting with 16 R bases (letters), and a mutation probability of $\frac{1}{4}$ (i.e. the probability of $R \rightarrow Y$ and $Y \rightarrow R$ is $\frac{1}{4}$ in each generation, for each base independently), what is the average number of R's out of the 16 after 1 generation? And after 3 generations? Justify your answer.

Using the formula $P_n = \frac{1-(1-2p)^n}{2}$ and taking $p = 1/4$ we get $P_1 = 1/4$ and $P_3 = 7/16$, which gives $16 * 1/4 = 4$ and 7 Y's, or 12 and 9 R's after 1 and 3 generations.

(*3 marks*)

(c) What is the "Principle of Optimality"?

Sub-optimal solution of sub-problem cannot be part of optimal solution of the original problem.

(*1 marks*)

(d) Explain why the BLOSUM50 matrix contains positive match scores for some entries not on the main diagonal.

See lecture notes on Log-odds.

(*2 marks*)

(e) Align the sequences **ATT** and **TA** using the Needleman&Wunsch global alignment algorithm. Use a match score of **5**, mismatch score of **-3** and a linear gap penalty of **-2**. Clearly indicate the traceback arrows inside the table of scores.

		A	T	T
	0	-2	-4	-6
T	-2	-3	3	1
A	-4	3	1	0

The alignment traceback goes from cell (3,4) to (2,3),(1,2) and (1,1). The alignment is,

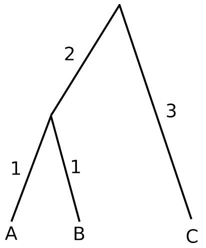
```
A T T
- T A
```

(*4 marks*)

(f) Use the UPGMA clustering method to build a phylogenetic tree for the taxa **A,B,C** from the distance matrix below. List the main steps explicitly.

	A	**B**	**C**
A	0		
B	2	0	
C	5	7	0

The smallest entry is 2, so we join A and B first at height 2/2 = 1. There are only 2 nodes left, so we simply compute the distance between AB and C as the average between the distances of AC and BC, (5+7)/2 = 6. We join them with a node at height 3 (6/2), giving the rooted tree below,



(*4 marks*)