Homework 3

Deadline: 11:59pm, March 29, 2013
Available points: 112. Perfect score: 100.

You will receive 10% extra credit points if you submit your answers as a typeset PDF (preferably using LaTeX, in which case you can also submit electronically your source code). There will be a 5% bonus for typewritten but not typesetted answers. Resources on how to use LaTeX are available on the course's website. **Do not submit Word documents, raw text, etc.** Make sure to generate and submit a PDF if you want to get the extra credit points. In this case you can submit your solutions electronically through `sakai.rutgers.edu`.

If you choose to submit handwritten answers and we are not able to read them, you will not be awarded any points for the part of the solution that is unreadable. Handwritten answer-sheets can be submitted to the instructor in class or to one of the TAs during office hours.

Try to be precise. Have in mind that you are trying to convince a very skeptical reader (and computer scientists are the worst kind...) that your answers are correct.

Each pair of students must write its solutions **independently from other teams**, i.e., without using common notes or worksheets with other students. Each pair of students need to submit only one copy of their solutions. You must indicate at the top of your homework who you worked with. You must also indicate any external sources you have used in the preparation of your solution. **Do not plagiarize online sources and in general make sure you do not violate any of the academic standards of the course, the department or of the university (the standards are available through the course's website)**.

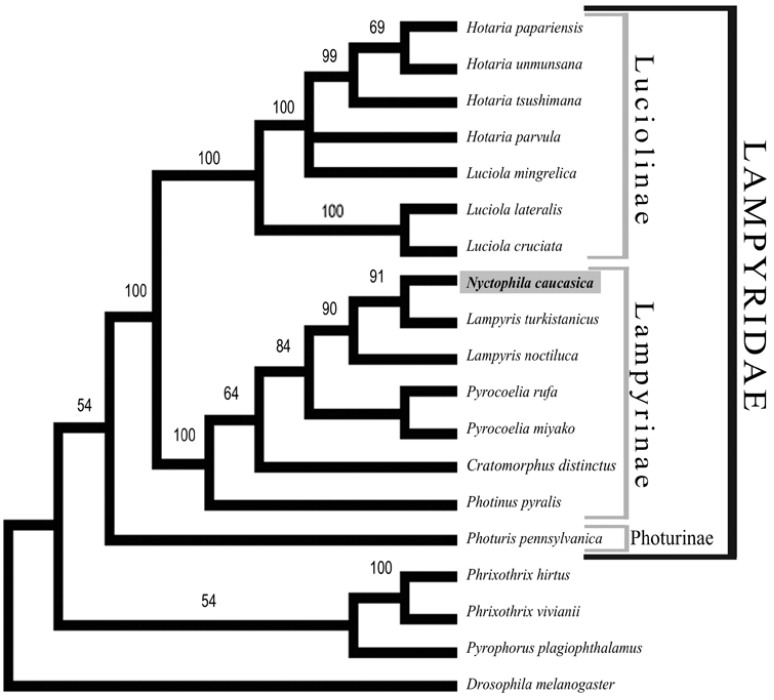

Figure 1: An example of a binary evolutionary tree.

Problem 1 (24 points): Consider that you have the evolutionary (binary) tree of n biological species, i.e., a tree where the organisms are placed at the leaves and internal nodes correspond to common ancestor species as in the previous Figure.

Suppose also that you have managed to sequence a particular gene across these species, which appears mutated among them. The gene has length k and is made out of letters from the following alphabet: $\{A, C, G, T\}$. We would like to identify the most probable version of this gene for the common ancestor of the n species given the available binary evolutionary tree.

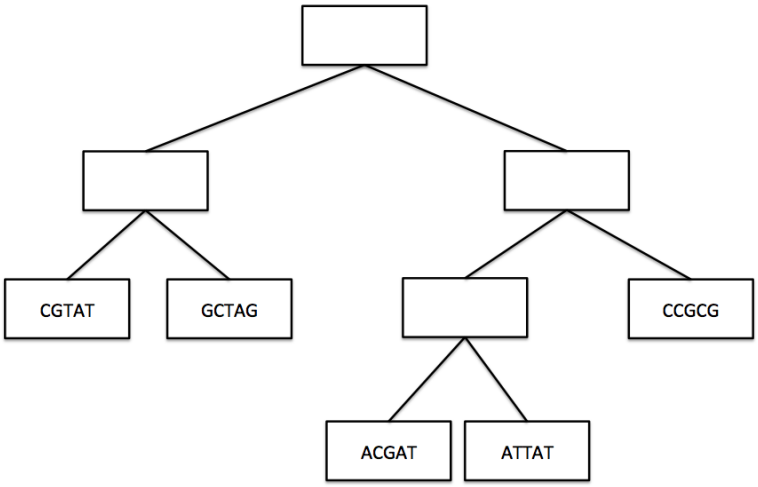

Figure 2: Example gene sequences of length 5 for five species and corresponding binary evolutionary tree.

The principle we are going to follow is that the most probable explanation for the ancestor genes corresponds to the one that minimizes mutations. A mutation occurs when the gene of the parent $g(p)$ and the gene of the child $g(c)$ have a different letter at the same position. For instance, the genes $g(p) = \{ACTGC\}$ and $g(c) = \{TCTAC\}$ exhibit two mutations, one in the first position from A to T and one in the fourth position from G to A. Consequently, we can assign a score for each subtree T that we want to maximize:

$$agreement(T) = \sum_{(p,c) \in E(T)} (\text{number of positions on which } g(p) \text{ and } g(v) \text{ agree})$$

A. Given the binary evolutionary tree for n species and genes with k symbols, provide an efficient algorithm for identifying the most probable ancestor genes. Argue about its correctness and running time.

B. Illustrate how the algorithm will work for Figure 2 and provide a possible explanation for the ancestor gene, as well as the corresponding agreement score.

Problem 2 (20 points): Consider a directed graph $G(V, E)$, where each node is associated with a positive integer p_u. Define then the payoff of a vertex as follows:

$$payoff[u] = \{\text{the maximum } p_v \text{ value among all nodes } v \text{ reachable from } u\}.$$

A. Provide a linear-time algorithm that can compute the payoff value for all the nodes of a directed acyclic graph. Remember that you can order the vertices of a directed acyclic graph.

B. Provide a linear-time algorithm that can compute the payoff value for all the nodes of a directed graph. Remember that a directed graph always has a hierarchical structure.

Problem 3 (20 points): A recent discovery led to identifying n articles written in the early 18^{th} century by different anonymous authors in the United Kingdom. These documents were discussing various subjects of interest at the time, such as monarchical absolutism. There were two political philosophies that were developing in the British scene at the time, the Whigs and the Tories. It is difficult to identify directly whether each article was written by a Whig or a Tory sympathizer, as the exact positions of the two competing ideologies at the time were evolving and they were not actually formal political parties. Nevertheless, we are able to identify x pairs of articles which take opposite positions on similar subjects and obviously belong to a different ideological group.

Provide a linear time algorithm as a function of n and x that determines whether it is possible to identify some of the articles as belonging to a single ideology and some of the articles as belonging to the opposite ideology (not necessary to identify which ideology, however). The algorithm should return this designation of articles into groups of similar ideology.

Problem 4 (24 points): Consider the problem of identifying handwritten characters, for instance identifying the amount of dollars on a handwritten bank check. Assume that we have available a graph $G(V, E)$. Each edge (u, v) of the graph carries a tag that corresponds to a character from a finite set of characters, e.g., $\{1, 2, \ldots, 9, 0\}$. If we follow a path on the graph, then the corresponding tags on the edges of the path define a sequence of digits.

Graph G represents prior knowledge about the type of numbers that appear in a specific application (e.g., 9s tend to appear frequently on prices of products and repetitively towards the end of a number). Furthermore, assume that every edge stores a probability $P(u, v)$ of going from u to v and identifying the corresponding digit. The sum of the probabilities of the edges leaving a vertex equals 1. The probability of a path from v_{init} is the product of the probabilities of its edges and equals the probability of a random process starting at v_{init} will follow this path, where the random process makes a probabilistic choice at each vertex of what edge to follow next based on the probabilities of the outgoing edges.

A. Given such a graph $G(V, E)$, an initial vertex v_{init} and a sequence of digits $N = \{d_1, d_2, \ldots, d_k\}$ from the finite set, provide an efficient algorithm that returns a path on $G(V, E)$ that begins at v_{init}, has edges that correspond to the sequence of digits in N. The algorithm should be able to detect that a path does not exist. What is the running time of the efficient solution? (Ignore the probabilities here.)

B. Compute the path with the highest probability of occurring that starts at v_{init} and corresponds to a specific sequence of digits $N = \{d_1, d_2, \ldots, d_k\}$. What is the running time of the efficient solution?

Problem 5 (24 points): The small town of Guryevsk is trying to compute an efficient route for their single snow plow truck that visits each street exactly once. All the streets in Guryevsk are two-way streets. The objective is to find whether it is possible for the snow plow to start from its station, go over all the streets exactly once and then return back to its station.

A. Show that the above problem has a solution if and only if there are no dead ends in Guryevsk and no n-way junctions, where n is odd, i.e., every intersection in Guryevsk has an even degree.

B. What are the conditions for the outline of Guryevsk's road network that will allow the existence of a solution to the following problem: Is there a way for the snow plow to start from its station, go over all the streets exactly once and then finish its path somewhere else?

C. What would the corresponding conditions for problem A be, if all the streets in Guryevsk are one-way streets.