Fall 2019 B503 Homework 4

Due date: 11:59 pm, Nov. 10   Lecturer: Qin Zhang

Your Name:________________   Your University ID:________________

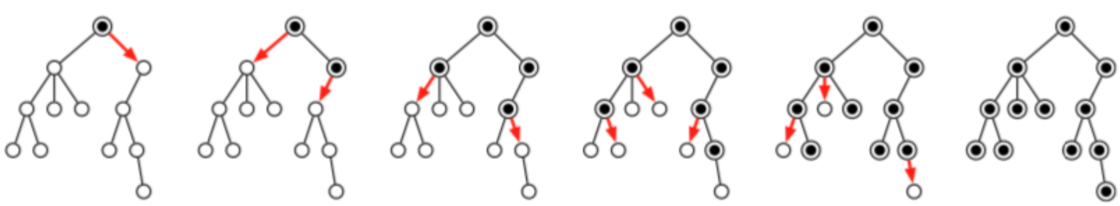
Instruction: Please submit your homework **before class on the due date, via Canvas**. Please add references to ALL the resources you have used for completing the assignment. You are allowed to discuss the assignment with other students, and if you do so, please list their names in the submission.

Due: 11:59 pm, Nov. 10

Total points: 70

Late Policy: No extensions or late homeworks will be granted, unless a request is made to the course instructor before due date and written documents are provided to support the reason for late submission.

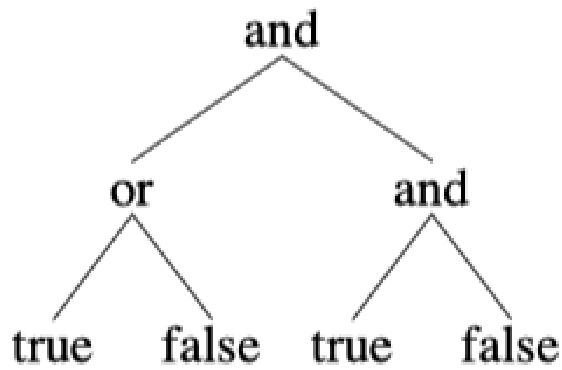
Problem 1 (10 points). Suppose we need to distribute a message to all the nodes in a rooted tree. Initially, only the root node knows the message. In a single round, any node that knows the message can forward it to at most one of its children. Describe and analyze an efficient algorithm to compute the minimum number of rounds required for the message to be delivered to every node.



A message being distributed through a tree in five rounds.

Problem 2 (10 points). At Aperture Bakeries, every cake comes with a binary boolean-valued tree indicating whether or not it is available. Each leaf in the tree has either a *true* or a *false* value. Each of the remaining nodes has exactly two children and is labeled either *and* or *or*; the value is the result of recursively applying the operator to the values of the children. One example is the following tree:

If the root of a tree evaluates to *false*, like the one above, the cake is a lie and you cannot have it. Any *true* cake is free for the taking. You may modify a tree to make it *true*; the



only thing you can do to change a tree is to turn a *false* leaf into a *true* leaf, or vice versa. This costs \$1 for each leaf you change. You can't alter the operators or the structure of the tree.

Describe an efficient algorithm to determine the minimum cost of a cake whose tree has n nodes, and analyze its running time.

Problem 3 (10 points). Consider two vertices, s and t, in a *directed acyclic graph* $G = (V, E)$. Give an efficient algorithm to determine whether the number of paths in G from s to t is odd or even. Analyze its running time in terms of number of vertices n and number of edges m.

Problem 4 (10 points). Recall that in the standard sequence alignment problem, you are given two strings X and Y , of length n and m, respectively, over a common alphabet Σ. The input also includes the penalty α_{gap} of inserting a gap and the penalty α_{ab} of (mis)matching each pair of characters a, b.

It often makes sense that the penalty for a gap of length 10, say, should be strictly smaller than 10 times the penalty of a single gap. (This reflects the possibility that a large chunk of a genome may get added or deleted via a single "mutation".)

To model this, consider a generalization of the sequence alignment problem in which, instead of a single gap penalty α_{gap} , the input includes parameters α_0, α_1 , and where the penalty of inserting exactly $k \geq 1$ gaps in a row is $\alpha_0+k\alpha_1$. (α_0 is larger than α_1) For example, the string AGT can be matched with ACCCGCCT with total penalty $(\alpha_0+3\alpha_1)+(\alpha_0+2\alpha_1) = 2\alpha_0 + 5\alpha_1$.

Design a dynamic programming algorithm that correctly solves this generalized sequence alignment problem and runs in $O(mn)$ time. Your algorithm should compute the value (i.e., minimum-possible total penalty) of an optimal alignment.

Problem 5 (10 points). In this problem, we consider a problem about clustering n points on the number line into k clusters. The input are distinct points $x_1, x_2 \ldots, x_n \in R$ in sorted order, and a parameter $k \leq n$. We would like to divide the n points into k disjoint non-empty clusters $S_1, \ldots, S_k$such that $\bigcup_{i=1}^k S_i = \{x_1, x_2 \ldots, x_n\}$ and all points in S_i are to the left of all points in S_{i+1} for $1 \leq i < k$, i.e., for any $y \in S_i, z \in S_{i+1}, y < z$.

The goal is to minimize the following objective function:

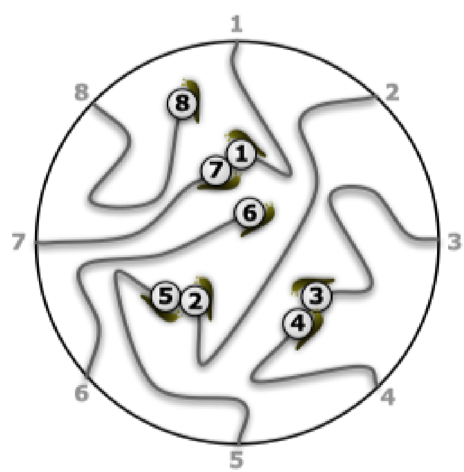
$$\sum_{i=1}^{k} [\max(S_i) - \min(S_i)]^2$$

Note that $\min(S_i)$ is the minimum element of S_i , i.e., the leftmost point of S_i. Similarly, $\max(S_i)$ is the maximum element of S_i, i.e., the rightmost point of S_i.

Design an $O(n^2k)$ time algorithm to find the optimal clustering using dynamic programming. Prove the correctness and analyze running time of your algorithm.

Problem 6 (10 points). Every year, as part of its annual meeting, the Antarctican Snail Lovers of Upper Glacierville hold a Round Table Mating Race. Several high-quality breeding snails are placed at the edge of a round table. The snails are numbered in order around the table from 1 to n. During the race, each snail wanders around the table, leaving a trail of slime behind it. The snails have been specially trained never to fall off the edge of the table or to cross a slime trail, even their own. If two snails meet, they are declared a breeding pair, removed from the table, and whisked away to a romantic hole in the ground to make little baby snails. Note that some snails may never find a mate, even if the race goes on forever.

For every pair of snails, the Antarctican SLUG race organizers have posted a monetary reward, to be paid to the owners if that pair of snails meets during the Mating Race. Specifically, there is a two-dimensional array $M[1..n, 1..n]$ posted on the wall behind the Round Table, where $M[i, j] = M[j, i]$ is the reward to be paid if snails i and j meet.



The end of a typical Antarctican SLUG race. Snails 6 and 8 never find mates.
The organizers must pay $M[3,4] + M[2,5] + M[1,7]$.

Describe and analyze an algorithm to compute the maximum total reward that the organizers could be forced to pay, given the array M as input.

Problem 7 (10 points). Arthur Dent has \$500 and 1000 hours to go from Cambridge, MA, to Berkeley, CA. He has a map of the States represented as a directed graph $G = (V, E)$. The vertices of the graph represent towns, and there is a directed edge $e = (A, B)$ from town A to town B if there is some means of public transportation connecting the two towns. Moreover, the edge is labeled with a pair (m_e, t_e), representing the cost $m_e \in \{0, 1, \dots\}$ in dollars of transportation from A to B and the time $t_e \in \{0, 1, \dots\}$ in hours that it takes to go from A to B. But m_e and t_e cannot both be 0.

Arthur is interested in finding a path from Cambridge to Berkeley that does not cost more than \$500 and does not take more than 1000 hours, while also minimizing the objective $5M^2 + 2T^2$, where M is the cost of the trip in dollars and T is the duration of the trip in hours.

(a) Due to his lack of knowledge in algorithms, he gave up on the idea of respecting the budget and time constraints. At least he thought he could efficiently find the path minimizing the objective $5M^2 + 2T^2$. He tried modifying Dijkstra's algorithm as follows: If an edge e was labeled (m_e, t_e), he assigned it a weight $w_e = 5m_e^2 + 2t_e^2$, and ran Dijkstra's algorithm on the resulting weighted directed graph. Show that Arthur's algorithm may return incorrect results,

(b) Now provide an algorithm that solves Arthur's original problem in time polynomial in number of vertices n and number of edges m. Your algorithm should find the path that minimizes the objective $5M^2 + 2T^2$, while at the same time respecting the constraints $M \leq 500$ and $T \leq 1000$. Please describe a linear time algorithm, and justify its correctness and running time. (Hint: M and T are upper bounded by a constant number.)