CS 170 Spring 2016 | Efficient Algorithms and Intractable Problems
Alessandro Chiesa and Umesh Vazirani | HW 8

Due March 14, 11:59pm

1. **(20 pts.) Sugar Water Enterprises** Your parents have just left for Costa Rica and have, surprisingly, left you in charge of their beverage company, Sugar Water Enterprises. Unfortunately, they failed to decide on a production plan before they left, and so you must do so now. Your company uses sugar, which costs 10 cents per gram, caffeine, which costs 20 cents per milligram, and water, which costs 1 cent per ounce to produce its beverages. There are two beverages that your company produces, Zap Energy and the signature Sugar Water. One case of Zap Energy is made with 5 grams of sugar, 1 milligram of caffeine and 8 ounces of water. One case of Sugar Water is made with 12 grams of sugar and 16 ounces of water. Your Sugar Water machine can't produce more than 80 Sugar Water cases in a day, and your Zap Energy machine can't produce more than 50 Zap Energy cases in a day. You have a budget of $b$$ that can be spent on materials. Zap Energy sells for 5$ per case, and Sugar water sells for 2$ per case. The goal is to maximize your profit in one day.

   (a) Create a linear program to represent this problem.

   (b) Find the dual of the linear program.

   (c) Find the optimal solution as a function of b (assume you can produce non-integer numbers of cases and can purchase items in non-integer quantities).

2. **(20 pts.) Zigzag sequences**

   (a) **(10 pts.)** A zig-zagging sequence is a sequence of numbers, $a_1, \ldots, a_k$ where $a_1 > a_2 < a_3 \cdots$ or $a_1 < a_2 > a_3 \cdots$: for example, $[3,2,4,3,5]$ is a zig-zagging sequence.

   Give a subproblem definition, recurrence, and base cases for your subproblem for a dynamic programming algorithm for finding the length of the longest zig-zagging subsequence in a sequence. Also, state how you would output a final answer.

   (Note that $[1,9,7,10,4]$ is a zig-zagging subsequence of $[1,5,9,8,7,10,4]$.)

   The running time is not needed, but the associated dynamic programming algorithm should be polynomial.

   (b) **(10 pts.)** Given two sequences, find the length of the longest common zig-zagging subsequence.

3. **(20 pts.) Reconstructing evolutionary trees by maximum parsimony**

Suppose we manage to sequence a particular gene across a whole bunch of different species. For concreteness, say there are n species, and the sequences are strings of length k over alphabet $\Sigma = \{A, C, G, T\}$. How can we use this information to reconstruct the evolutionary history of these species? Evolutionary history is commonly represented by a tree whose leaves are the different species, whose root is their common ancestor, and whose internal branches represent speciation events (that is, moments when a new species broke off from an existing one). Thus we need to find the following:

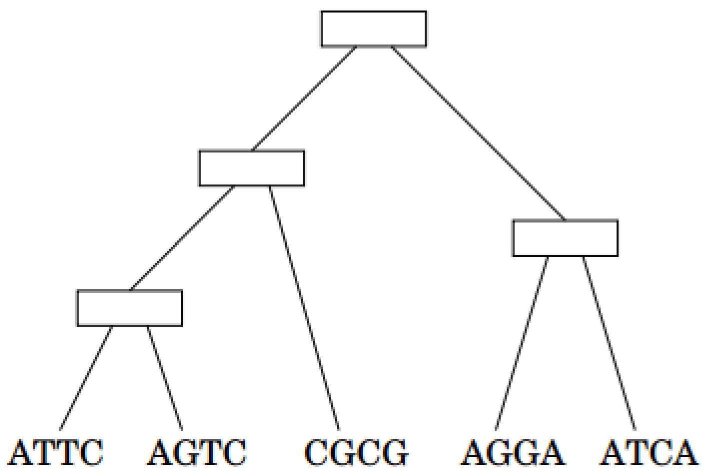
- An evolutionary tree with the given species at the leaves.

- For each internal node, a string of length k: the gene sequence for that particular ancestor.

For each possible tree T, annotated with sequences $s(u) \in \Sigma^k$ at each of its nodes u, we can assign a score based on the principle of parsimony: fewer mutations are more likely.

$$score(T) = \sum_{(u,v)\in E} (\text{number of positions on which } s(u) \text{ and } s(v) \text{ disagree}).$$

Finding the lowest-score tree is a difficult problem. Here we will consider just a small part of it: suppose we know the structure of the tree, and we want to fill in the sequences $s(u)$ of the internal nodes u. Heres an example with $k = 4$ and $n = 5$:



(a) In this particular example, there are several maximum parsimony reconstructions of the internal node sequences. Find one of them.

(b) Give an efficient (in terms of n and k) algorithm for this task. (*Hint:* Even though the sequences might be long, you can do just one position at a time.)

4. (20 pts.) Airline scheduling (vertex disjoint path cover)

(a) Given a directed acyclic graph $G = (V,E)$, find the minimal number of vertex-disjoint paths that together cover all vertices (i.e. each vertex appears in exactly one path).
(*Hint #1:* Use max-flow.
Hint #2: $|V| = (\text{number of paths}) + (\text{total number of edges in paths})$.)

(b) You run an airline. There are n airports, with a flight from airport i to airport j taking d_{ij} time. Based on consumer demand, there are m pre-scheduled flights that you must make: each flight is from airport a_i to airport b_i, and must depart at exactly time t_i.
What is the minimal number of planes you need in order to be able to handle all m flights? (Planes can fly between airports outside of the m required flights, these extra flights still take the same amount of time.)

5. (20 pts.) Verifying a max-flow

Suppose someone presents you with a solution to a max-flow problem on some network. Give a *linear* time algorithm to determine whether the solution does indeed give a maximum flow.

6. (5 optional bonus pts.) Star-shaped polygons in 2D

Consider a polygon in two dimensions. The polygon is specified by an ordered list $p_1 = (x_1, y_1), \ldots, p_n = (x_n, y_n)$ where the n line segments between p_i and p_{i+1} for $i \in \{1, 2, \ldots, n-1\}$ and p_n,p_0 do not intersect except at endpoints. Since a polygon does not cross itself, the boundary of the polygon splits the plane into two regions. Each segment is adjacent to both regions and, accordingly, has two sides. Suppose that for each segment we are given as part of the input which side corresponds to the interior of the polygon.

Call a polygon *star-shaped* if there a point inside of the polygon from which all points on the boundary of the polygon can be seen. That is, the polygon is star-shaped iff there is a point x inside the polygon such that for every y on the boundary of the polygon, the segment xy only intersects the polygon at y.

Write a linear program that can be used to identify whether or not a polygon is star-shaped.