

Algorithms in Bioinformatics

close-book midterm exam

December 11, 2002

Instructions

1. **DO NOT** write any of your identification information (including name, student ID number, etc.) on your answer sheets. Instead, please put the *exam ID* assigned to you on **each page** of your answer sheets.
2. Throughout the exam, let X denote the 7-bit binary string representing your exam ID. For example, if your exam ID is 5, then your X is the binary string 0000101. Also, let your string S be obtained from X as follows. For each $i = 1, 2, \dots, 7$, let $S[2i - 1] = 2$ and $S[2i] = X[i]$. For example, if your X is 0000101, then your S is 20202020212021.
3. Cheating will be most seriously punished. Any dishonest attempt in this exam implies an F as your final grade.
4. You are welcome to use anything that we have taught in our lectures as a subroutine to your answers.

Problem 1 (15 points)

- (5 points) Write down your strings X and S . (Do “triple check” your answers, since at least half of the following problems depend on your answers to this subproblem.)
- (5 points) Give the Z values for your binary strings X and S .
- (5 points) We taught a simplified $O(|S|)$ -time algorithm for computing the Z values of any input string S . Unfortunately, that (over-simplified) algorithm is buggy. Please give a correct $O(|S|)$ -time algorithm that computes all Z values of S . (Hint: see item 4 of the instruction.)

Problem 2 (20 points)

Give a suffix tree (in linear-space format) for your length-14 string S . Your answer has to contain a correct label $[x, y]$ for each edge as well as the correct suffix link for each internal node. Moreover, for convenience of grading your answer, the descending edges of each internal (i.e., non-leaf) node have to obey the alphabetical order, i.e., 0, 1, 2 from left to right.

Your grade depends only on the final picture of your suffix tree. The process you obtain your answer does not count.

Problem 3 (15 points)

You are asked to design a linear-time algorithm for computing a longest common substring for k strings. Specifically, given k binary strings S_1, S_2, \dots, S_k , your algorithm should run in $O(|S_1| + |S_2| + \dots + |S_k|)$ time and outputs a longest string C that occurs in each S_i with $1 \leq i \leq k$.

Problem 4 (20 points)

Let S be the string obtained from X as explained in Problem 2.

- (10 points) Give the dynamic-programming table for computing the local alignment of your S and the string 210210. The scores of “match”, “ordinary mismatch”, and “gapped mismatch” are +4, -2, and -1, respectively. For example, $\text{score}(2, 2) = 4$, $\text{score}(1, 0) = -2$, $\text{score}(-, 1) = -1$. You should also write down an optimal local alignment obtained from your table.
- (10 points) Give the dynamic-programming table for computing the (edit) distance-3 matching using you S and the pattern string $P = 012012$. You should also write down the resulting output indices obtained from your table.

Problem 5 (15 points)

Give a linear-time algorithm for computing a minima tree for any input string of numbers. Specifically, given a string S of numbers, your algorithm should run in $O(|S|)$ time and output a minima tree for S .

Problem 6 (15 points)

Explain how to compute in polynomial time an optimal lifted assignment for a rooted tree T with strings on its leaves.