T-106.5400 String Algorithms
Exam May 19, 2014

Examiner: Travis Gagie

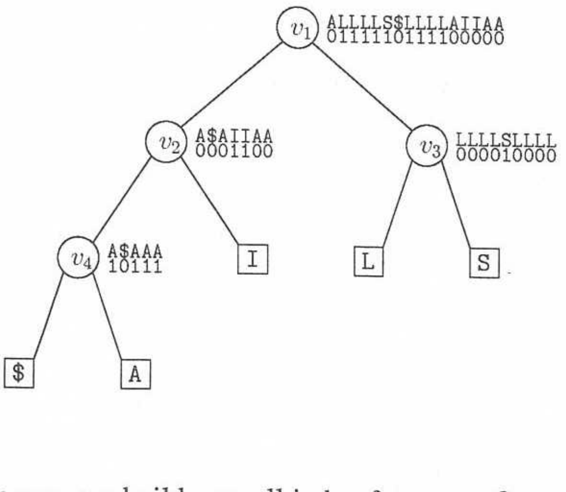
No written material is allowed in this exam. Submit at least one answer sheet, even if an empty one! Write on *each* answer sheet you submit the code of the course, the date, your name, and your student ID number. This exam has 5 questions (continued on the back of the page) each worth 20%; try to answer them all.

---

1. Let $S_1 = \texttt{ILLALLASILLALLA\$}$ and assume \$ is lexicographically smaller than all other characters. For the following questions, you don't need to show how you arrive at your solutions:

   - Build the suffix tree for S_1.
   - Compute the LZ77 parse of S_1.
   - Build the suffix array for S_1.
   - Build a Huffman code for the distribution of characters in S_1.
   - Apply move-to-front coding to S_1, just writing the numbers in decimal, starting with the list \$, A, I, L, S.

2. Below are a wavelet tree for the BWT of S_1 and the function $C(a)$ that returns the partial sums of the characters' frequencies. Suppose each node v supports the queries $v.\mathsf{rank}_0(i)$ and $v.\mathsf{rank}_1(i)$ on the binary sequence it stores. Considering the tree as an FM-index, list the rank queries used to count the number of occurrences of LAL in S_1.



a	\$	A	I	L	S
$C(a)$	0	1	5	7	15

3. How can you build a small index for a text S_2 such that later, given a pattern P and an integer k, in $\mathcal{O}(|P|)$ time you can find the longest substring of P that occurs in S_2 at least k times? (If you can't get $\mathcal{O}(|P|)$ time, $\mathcal{O}(|P|^2)$ time is ok.)

   Hint: Use a suffix tree of S_2 whose internal nodes are labelled with the numbers of leaves in their subtrees.