Homework 4. Due November 15, 9:59PM.

CS181: Fall 2021

GUIDELINES:

- Upload your assignments to Gradescope by 9:59 PM.
- Follow the instructions mentioned on the course webpage for uploading to Gradescope very carefully (including starting each problem on a new page and matching the pages with the assignments); this makes it easy and smooth for everyone. As the guidelines are simple enough, bad uploads will not be graded.
- You may use results proved in class without proofs as long as you state them clearly.
- Most importantly, make sure you adhere to the policies for academic honesty set out on the course webpage. The policies will be enforced strictly. Homework is a stepping stone for exams; keep in mind that reasonable partial credit will be awarded and trying the problems will help you a lot for the exams.
- Note that we have a **modified grading scheme for this assignment**: A sincere attempt will get you 100% of the credit and a reasonable attempt will get you 50% for each problem. Nevertheless, please attempt the problems honestly and write down the solutions the best way you can - this is really the most helpful way to flex your neurons in preparation for the exam.
- All problem numbers correspond to our text 'Introduction to Theory of Computation' by Boaz Barak. So, exercise a.b refers to Chapter a, exercise b.

Remark: To receive full credit, first explain at a high-level how the machine works (e.g., how we described the machine for Palindrome/Minority of 0's in class). You should then provide a few more details about each step as to how you implement them **but** you don't have to dig in and write down all the individual transitions, etc. For example, it would be ok to say "Scan right until you reach end and enter state $****$". But saying "Pair up 0s and 1s until none are left" for the PALINDROME problem would not a valid solution. If you are not sure about this, contact the TAs or ask on edstem for clarification. (The idea is to make it easy for you to write down the solution without cumbersome notation and make it easy for the readers to grade the solutions.)

1. Design a TM that recognizes $L = \{x :$ number of 1's in x is at least twice the number of 0's.$\}$. So for instance $110, 1110, 011011 \in L$, whereas $10, 11100 \notin L$. Do not use HOC here but describe the TM in pseudocode as we did in class for Maj. [1 point] [1 point]

2. Exercise 7.4. You can assume that at the start of computation, the first tape is loaded with the input (as in class) and the second tape is empty. The output of the machine will be the output from the first tape only. Do not use HOC here but describe the TM in pseudocode as we did in class for Maj. [1 point]

   [Hint: Try to use a special symbol (say, #) to concatenate the 'two tapes' and use special marked symbols to note the position of the two heads.]

3. Define a function $DecbyOne : \{0,1\}^* \to \{0,1\}^*$ that takes the binary representation of an integer as input, and returns the binary representation of the number minus 1. The answer should have the same the number of bits as the input (so you may end up padding with zeros if needed). So for instance, $Decbyone(1100) = 1001$, $Decbyone(0010) = 0010$, $Decbyone(1000) = 0111$. Do not use HOC here but describe the TM in pseudocode as we did in class for Maj. [1 point]

4. This problem essentially shows that TMs can actually implement indexable arrays.

   Define a function $Ind : \{0,1\}^* \circ \{\#\} \circ \{0,1\}^* \to \{0,1\}$. That is the input is of the form $i\#x$ where $i \in \{0,1\}^*$, $x \in \{0,1\}^*$. We interpret i as the binary representation of an integer and $Ind(i\#x) = x[i]$. For example, $Ind(0\#101010) = 1$ as the bit in the 0'th position of x is 1. Similarly, $Ind(1\#101010) = 0$, $Ind(11\#101010) = 0$ as the bit in the third position of x is 1. (Remember indexing starts from 0.)

   Give a TM that computes Ind. That is, for instance, when the tape is loaded with $i\#x$, the TM ends with $x[i]$ on its tape. You can assume without loss of generality that the integer whose binary representation is i is less than the length of x (i.e., don't worry about border cases). Do not use HOC here but describe the TM in pseudocode as we did in class for Maj. [1 point]

   [Hint: You can use the idea behind Decbyone as a building ingredient. You can even give a two tape TM if that simplifies your pseudocode. Imagine keeping a separate head at the start of x, how many times do you have to move it to the right to get the right answer?]

Practice problem. Do not submit.

1. Show that computable functions are closed under the "concatenation" operation: If $F, G : \{0,1\}^* \to \{0,1\}$ are two functions, define a new function $H : \{0,1\}^* \to \{0,1\}$ with $H(x) = 1$ if x can be written as $x = x_1 \circ x_2$, $F(x_1) = G(x_2) = 1$; and $H(x) = 0$ if there is no way to break up x as such. Show that if F, G are computable, then so is H. [1 point]

   [Hint: You can give a multi-tape TM for H. That is, you can use two-tapes or three or even four tapes. You don't have to describe how these multi-tape TMs are equivalent to one-tape. Make sure, you give sufficient details to describe how to implement this high-level idea.]