Eric Roberts
CS 54N

Handout #13
October 19, 2016

Calendar for Week #5

Monday, October 24

Monday's class introduces one of the most amazing ideas in computer science, which is that some well-defined functions can nonetheless be uncomputable, at least by any finite mechanical process. Several mathematicians proved this result independently during the 1930s, but the proof that has had the most profound influence on modern computer science is the one published by Alan Turing in 1936. In this lecture, I will follow the organization of Chapter 9 and prove the same result using a more modern formulation that grows out of the Busy Beaver problem.

Reading: Chapter 9. Before you come to class on Monday, you should try (without looking at solutions on the web) to come up with a three-state Turing machine program that writes out the largest possible number of **1**s. Most students can, with an hour or two of work, come up with a three-state machine that produces the maximal value of six **1**s.

Wednesday, October 26

The idea that some functions are not computable by any mechanical process leads directly to the question of how much effort is required to evaluate the functions that are computable. This topic is a rich area of theoretical computer science research and includes the **P=NP** question, which is one of the most famous open problems in mathematics—so famous that it carries a $1,000,000 prize for its solution. On Wednesday, I'll introduce the **P=NP** question and talk about why it is so important.

Reading: Chapter 10. I recommend that you skim—as opposed to read—section 10.4 on Cook's Theorem. The example is almost certainly too complicated, and I need to think of a less confusing way to introduce this idea.