

# Modern Complexity Theory Homework One

The aim of problem set is to help you to test your understanding of Turing Machines.

- **Collaboration:** You can collaborate with other students that are currently enrolled in this course in brainstorming and thinking through approaches to solutions but you should write the solutions on your own and cannot share them with other students.
- **Owning your solution:** Always make sure that you “own” your solutions to this other problem sets. That is, you should always first grapple with the problems on your own, and even if you participate in brainstorming sessions, make sure that you completely understand the ideas and details underlying the solution. This is in your interest as it ensures you have a solid understanding of the course material, and will help in the midterms and final. Getting 80% of the problem set questions right on your own will be much better to both your understanding than getting 100% of the questions through gathering hints from others without true understanding.
- **Serious violations:** Sharing questions or solutions with anyone outside this course, including posting on outside websites, is a violation of the honor code policy. Collaborating with anyone except students currently taking this course or using material from past years from this or other courses is a violation of the honor code policy.
- **Submission Format:** The submitted PDF should be typed and in the same format and pagination as ours. Please include the text of the problems and write **Solution X:** before your solution. Please mark in gradescope the pages where the solution to each question appears. Points will be deducted if you submit in a different format.

**By writing my name here I affirm that I am aware of all policies and abided by them while working on this problem set:**

**Your name:** (Write name and emailid)

**Collaborators:** (List here names of anyone you discussed problems or ideas for solutions with)

## Questions

For a non bonus question, you can always simply write “**I don’t know**” and you will get 15 percent of the credit for this problem.

If you are stuck on this problem set, you can use this discussion board to send a private message to all instructors under the **e-office-hours** folder.

This problem set has a total of **50 points** and **5 bonus points**. A grade of 50 or more on this problem set is considered a perfect grade.

### 0.1 Short Problems

Following are short problems worth 4 points each.

**Question 1.1:** For a Turing Machine, can the input alphabet  $\Sigma$  be equal to the tape alphabet  $\Gamma$ ?

**Solution 1.1:**

**Question 1.2:** Can there be a Turing Machine that recognizes a non trivial language (i.e. not the empty language and not  $\{0, 1\}^*$ ) with a single state? Why/why not?

**Solution 1.2:**

**Question 1.3:** Consider a Turing Machine with 3 tapes such that it can read, write and move the head on all the tapes simultaneously. Write the formal specification (7-tuple) for such a Turing Machine.

**Solution 1.3:**

**Question 1.4:** Give an example of a language that can be recognized by a Turing Machine but not by a Context Free Grammar.

**Solution 1.4:**

**Question 1.5:** Consider a variant of a push-down automata where the stack is replaced by a FIFO queue. Does this change give it more power compared to usual push down automata in terms of the languages that can be recognized?

**Solution 1.5:**

### 0.2 Conceptual Problems

**Question 2.1 (3+7 Points)**

- (a) Construct a single tape Turing machine that given a number as input computes its quotient with 2. Assume the input is present on the tape in binary.
- (b) Construct a single tape Turing machine that reverses its input. Assume the input is present on the tape in binary (e.g., produces “0010111” from “1110100”).

**Question 2.2 (10 Points):** Consider a Turing Machine with an infinite 2-D tape. The head can now move not only left and right but also up and down. The input is initially written to the right of the head position.

- (a) Write a detailed formal specification for this Turing Machine. How will the transition function change? What is a configuration?
- (b) Does the set of languages recognized by such a 2D Turing Machine differ from that of the 3-tape Turing machine mentioned in Question 1.3? If so, give an example of a language that is recognized by one but not the other. If not, why? Would there be some sort of trade off in such a case?

**Solution 2.2:**

**Question 2.3 (10 Points):**

- (a) What is a language that is recursively enumerable but not recursive?
- (b) What is a language that is recursive but not recursively enumerable?
- (c) Read about Enumerators from Pg. 180 of "Introduction to the Theory of Computation, 3rd Edition by Michael Sipser". Read and understand the proof of Theorem 3.21 on the following page. (Write "I certify that I fully read the section on Enumerators and the proof of Theorem 3.21".)
- (d) Using your understanding of Enumerators and Theorem 3.21 from the previous question, show that a language is recursive if and only if there is an Enumerator that enumerates the strings of the language in a length increasing fashion.

**Solution 2.3:**

**Bonus Problem (5 Points):**

Construct a single tape Turing machine that adds two numbers written in binary. (Assume that the numbers are separated by a special symbol "+" that belongs to the external alphabet of the TM.

**Bonus Solution:**