Homework 8

You may collaborate with *one or two* other students on the graded homework if you wish, or work alone. Collaboration must be true collaboration however, which means that the work put into each problem should be roughly equal and all parties should come away understanding the solution.

The ungraded homework may be done in whatever way works best for you. There are no rules regarding collaboration. The point of ungraded homework is to develop your abilities and prepare you for the quiz. Solutions will be provided, but they should be consulted only when you need a hint and/or afterward to compare and contrast your solution with mine.

Graded Homework

Completion of these tasks by the time listed on the assignment is worth approximately 1-2% of your course grade. No late homework will be accepted.

1. I will soon place a "Homework quiz" on Canvas for you to complete. It will be untimed and you can take it as many times as you wish. You will be able to see your score after each submission, and the highest will be kept as your score.
2. I will soon place a programming problem on Mimir for you to complete.

Ungraded Homework

(1) Write a reduction from `median` (the problem of finding the middle integer in a list of integers) to `sort` (the problem of sorting a list of integers). What connection does this reduction establish between the two problems (express as an implication)? What is the contrapositive of this implication?

(2) A famous result says that there is no algorithm that can take as input another algorithm *M* and its input *x* and, in every case, determine whether *M* will eventually halt while processing *x*. If we wanted to show that some problem *A* was also not always computable, which of the following two reductions would do so, given that HaltSolver is not computable? Explain why you chose the one you did and write the logical implications the reduction establishes.

```
HaltSolver(M,w):                              ASolver(a_instance):
   // convert <M,w> into a_instance              // convert a_instance int <M,w>
   a_answer = ASolver(a_instance)                halt_answer = HaltSolver(M,w)
   // convert a_answer into halt_answer          // convert halt_answer into a_answer
   return halt_answer                            return a_answer
```

(3) Write a reduction from `min` (the problem of finding the smallest integer in a list of integers) to `max` (the problem of finding the largest integer in a list of integers). What connection does this reduction establish between the two problems (express as an implication)? What is the contrapositive of this implication? Hint: You may use a loop and you may call `remove` on an element to remove it. It's okay to call the function you are reducing to multiple times... It still shows how to solve one problem by using the solution to another.

(4) Let *M* be an algorithm that takes an input *x* and always outputs either *true* or *false*. It is known that there is no algorithm *Any(M)* that returns true if *M* outputs *true* for at least one of its possible inputs and *false* if *M* never outputs *true*. In other words there is no algorithm that can decide whether *M* accepts *any* input at all.

Show that because *Any* is not computable *Equal*(M_1, M_2) is also not computable, where *Equal* returns true if M_1 and M_2 output *true* for the exact same sets of inputs. To do this, reduce from *Any* to *Equal*. Hint: As the first line of your reduction write "Let M_2 be the algorithm that always outputs *false*".

(5) Read Sections I.1 and I.2 and do Exercises 1.13, 1.16, 1.20, 1.22 from *Theory of Computation*. Definitions 1.7 and 1.8 are rather technical and may be ignored.

Ungraded homework solutions

Study these only after completing the homework or after struggling with it for a while.

[Ungraded solutions]