The homework is due in class on Thursday, September 22. If you can’t make it to class, drop it in my mailbox in the MacLean Hall mailroom.

Important Note: These questions were assigned as homework questions in a previous offering. Please do not view those solutions. Also please see the first day handout for what is acceptable in terms of collaborating with other students.

1. Define a TM $M$ to be oblivious if its head movements do not depend on the input but only on the input length. That is, $M$ is oblivious if for every input $x \in \{0, 1\}^*$ and integer $i \geq 0$, the location of each of $M$’s heads at the $i$’th step of execution on input $x$ is only a function of $|x|$ and $i$. Show that for every time-constructible function $T$, if $L \in \text{DTIME}(T(n))$ then there is an oblivious TM that decides $L$ in $O(T(n)^2)$ time. Furthermore, show that there is such a TM that uses only two tapes: one input tape and one work/output tape. Hint: Modify the method in Claim 1.6. (This problem is Exercise 1.5 in the text.) (3 points)

2. Let $AEL : \{0, 1\}^* \rightarrow \{0, 1\}$ be the function defined as follows: $AEL(\alpha) = 1$ if $M_\alpha$ outputs 1 on any string $x \in \{0, 1\}^*$ such that $|x|$ is even, and 0 on any string $x \in \{0, 1\}^*$ such that $|x|$ is odd; $AEL(\alpha) = 0$ otherwise. Recall that $M_\alpha$ is the Turing machine encoded by $\alpha$. Show that there is no Turing machine for computing $AEL$. Hint: Do a reduction from the HALT function. This reduction is quite similar to the reductions we did in class, for the Hello-World and Accepts-All-Strings functions. (3 points)

3. This problem is a slight variant of Exercise 1.10 in the text. Consider the following simple programming language. It has a single infinite array $A$ of elements in $\{0, 1, 2\}$ (initialized to 2) and a single integer variable $i$. A program in this language contains a sequence of lines of the following form:

```
label: If A[i] equals $\sigma$ then cmds
```

where $\sigma \in \{0, 1, \square\}$ and $cmds$ is a list of one or more of the following commands:

1. Set $A[i]$ to $\tau$ where $\tau \in \{0, 1, \square\}$,
2. Goto label,
3. Increment $i$ by 1,
4. Decrement $i$ by 1, and
5. Output $b$ and halt, where $b \in \{0, 1\}$.

A program is executed on an input $x \in \{0, 1\}^n$ by placing the $i$’th bit of $x$ in $A[i]$, initializing $i$ to 1 (the first index in array $A$), and then running the program using the obvious semantics. $^2$

---

$^1$AEL abbreviates Accepts-Even-Lengths.

$^2$The programming language is a bit like assembly language.
Prove that if a function $f : \{0,1\}^* \to \{0,1\}$ is computable by a program in this language, then $f$ is also computable by a Turing Machine. You should do this by describing a Turing Machine that simulates the program. It is enough to do so at a high level, like in the proofs of Claims 1.5 and 1.6.

If the program computes $f$ in $T(n)$ time, how can we bound the running time of the corresponding TM? (4 points)