Comments on Homework 8

Note that these are comments on the class’s solutions, and not intended to be sample solutions.

1. (5 pts) Problem 14.14 on page 411. (Show that if $L$ can be recognized by a multi-tape TM with time-complexity $f$, then $L$ can be recognized by a one-tape TM with time-complexity $O(f^2)$.)

Many groups did well on this one. However, I do have a couple of comments.

- Try to reserve $n$ for the length of the input string. Use $i$, $j$, $k$, $\ell$, $m$ for other quantities.
- Some answers read as if they were lower bounds, saying “at least” so much time is needed, or “in the worst case ...” Your answers should be showing an upper bound – the simulation takes no more than a certain amount of time.
- If the multi-tape TM has $k$ tapes, then most groups had each symbol of the 1-tape TM represent a $k$-fold product of symbols from the multi-tape machine. It is possible to “list” the $k$ tapes one after the other (separated by an appropriate separator like “#”), but then you have to account for the extra time needed to shift everything down as the multi-tape TM uses more of its tapes.

A similar solution could use the function $\tau_T(n)$ (the maximum running time of $T$ on strings of size $n$) instead of $f(n)$, assuming it is computable in time $c(f(n))$. If one doesn’t care what the recognizing TM does on strings that are not valid encodings, then $\tau_T(n)$ doesn’t even have to be quickly computable.

2. (5 pts) Problem 14.16 on page 411. (Show that each solvable decision problem has an encoding such that the corresponding language can be recognized by a TM with linear time complexity.)

The common approach here was to encode instances in two ways, one way for the “yes” instances and another for the “no” instances. Although OK, this was not the idea I was trying to get across. The encoding can be padded so that it is long enough to have the deciding TM run in linear time. Here is a sketch of one way to do this.
Consider any (reasonable) encoding of the decision problem, so each instance $I$ has the encoding $e(I)$, and let $T$ be any TM recognizing the language of encoded “yes” instances. Let $f(x)$ be the time spent by $T$ when running on input $x$.

Now consider a modified encoding $e'(I) = e(I)\#1^{f(e(I))}$. This can be computed by first computing the encoding $e(I)$ and then simulating $T$ on the encoding.

A simple modification of $T$ that first copies everything after the “#” to a new tape and then runs $T$ on $e(I)$, stepping through the $1^{f(e(I))}$ to check that it is actually the number of steps (in unary) needed by $T$ to decide $x$. This modification runs in linear time, and decides the language of properly encoded “yes” instances of the decision problem.

3. (5 pts) Problem 14.26 on page 412. (Show that if input tape is “read only” and does not count as “space” then the palindrome and balanced parentheses languages can be decided in using space $1 + \lceil \log_2(n + 1) \rceil$.)

Most groups figured out how to use a counter to keep track of the number of unmatched “(” for the balanced parentheses problem, and how to keep one counter (or several counters on different tapes) for the palindrome problem. You could also keep several counters on one tape using “cross-product” symbols.