HW 4 solutions

JC

March 8, 2003

1. The first two parts are exercises in Euclid's algorithm.

$$\begin{aligned} 105 &= 6 \times 16 + 9 \\ 16 &= 1 \times 9 + 7 \\ 9 &= 1 \times 7 + 2 \\ 7 &= 3 \times 2 + 1 \end{aligned}$$

Tracing back

$$1 = 7 - 3 \times 2 = 4 \times 7 - 3 \times 9 = 4 \times 16 - 7 \times 9 = 46 \times 16 - 7 \times 105$$

so that the inverse class is $46 + 105\mathbb{Z}$.

$$\begin{aligned} 962829 &= 11 \times 80845 + 73534 \\ 80845 &= 1 \times 73534 + 7311 \\ 73534 &= 10 \times 7311 + 424 \\ 7311 &= 17 \times 424 + 103 \\ 424 &= 4 \times 103 + 12 \\ 103 &= 8 \times 12 + 7 \\ 12 &= 1 \times 7 + 5 \\ 7 &= 1 \times 5 + 2 \\ 5 &= 2 \times 2 + 1 \end{aligned}$$

So now

$$1 = 5 - 2 \times 2 = 3 \times 5 - 2 \times 7 = 3 \times 12 - 5 \times 7 = 43 \times 12 - 5 \times 103$$

$$= 43 \times 424 - 177 \times 103 = 3052 \times 424 - 177 \times 7311 = 3052 \times 73534 - 30697 \times 7311$$

$$= 33749 \times 73534 - 30697 \times 80845 = 33749 \times 962829 - 401936 \times 80845$$

So the inverse class is $-401936 + 962829\mathbb{Z} = 560893 + 962829\mathbb{Z}$.

For the third part let $n = p_1^{n_1} \dots p_k^{n_k}$ where the p_i are distinct primes and $n_i > 0$. Every class has the form $a + n\mathbb{Z}$ where $0 \le a < n$, and so the question is how many such a have the property that n divides some power of a. It is easy to see that n divides some power of a if and only if all the p_i divide a, and that in turn all the p_i divide a if and only if $m = p_1 \dots p_k$ divides a. So now we just need to find out how many a are multiples of m, and the answer is $n/m = p_1^{n_1-1} \dots p_k^{n_k-1}$.

2. If n is composite, say $n = ab$, then it is impossible that both $a > \sqrt{n}$ and $b > \sqrt{n}$. So n has a factor which is at most $\sqrt{n}$ and we only need to check for factors up to $\sqrt{n}$.

   For the second part we notice that if $n = p^2$ for p prime then both the algorithms run for p steps before finding that n is composite. The size of p^2 is at most $2\ln_2 p + 1$. Since there are infinitely many primes we get infinitely many sizes s such that a number of size s takes more than $2^{(s-1)/2}$ steps; the function $2^{(s-1)/2}$ grows faster than any polynomial function of s, so neither of the algorithms runs in polynomial time.

   IMPORTANT: This argument may seem roundabout but I had to do something like this; the algorithm only takes 2 steps on any even n, that is to say it can run very fast even on a very large n. The point is that there are infinitely many n where it runs slowly.

3. There are infinitely many correct answers to this question. Here is one. I have given the states suggestive names. Upper case L and R correspond to moving left or right, lower case l and r correspond to being to the right of cell 0 or the left of cell 0. C stands for a state where we are "carrying a 1". A is for a state where addition is performed, E is for a state where we erase. I stands for "initial" and F for "final", these are special states we use at the beginning and end of the computation.

   The machine starts by computing the least significant digit and writing it. On each subsequent pass it erases one digit of the input. When all input is erased it either stops immediately or goes back to write a final 1.

   I have given a fairly explicit description of the behaviour of the machine. Some combinations of states and symbols never occur and I have not specified what the machine is to do. The machine leaves the symbol in a cell unchanged unless I say otherwise, and stays in the same state unless I say otherwise.

   States: start, stop, IRr, IA, Lr1, Lr0C, Ll1, Ll0C, Rl, RlC, ZRr, ZRrC, Fr, Fl, Rr, RrC, E, EC, A, AC.

   As usual the machine is started in state start with the head on cell 0.

   Start: move right and enter state IRr.

   IRr: on 0 or 1, move right. On blank move left and enter state IA.

IA: on 0 enter state Lr1 and move left, on 1 enter state Lr0C and move left.

Lr1: on 0 or 1 move left, on blank enter state Ll1 and move left.

Lr0C: on 0 or 1 move left, on blank enter state Ll0C and move left.

Ll1: on 0 or 1 move left. On blank write 1, move right, enter state Rl.

Ll0C: on 0 or 1 move left. On blank write 0, move right, enter state RlC.

Rl: on 0 or 1 move right. On blank enter state ZRr and move right.

RlC: on 0 or 1 move right. On blank enter state ZRrC.

ZRr: on 0 or 1 move right and enter state Rr. On blank enter state stop.

ZRrC: on 0 or 1 move right and enter state RrC. On blank move left and enter state Fr.

Fr: move left and enter state Fl.

Fl: on 0 or 1 move left, on blank write 1, move left, enter state stop.

Rr: on 0 or 1 move right, on blank move left and enter state E.

RrC: on 0 or 1 move right, on blank move left and enter state EC.

E: write blank, move left, enter state A.

EC: write blank, move left, enter state AC.

A: on 0 move left and enter state Lr0, on 1 move left and enter state Lr1.

AC: on 0 move left and enter state Lr1, on 1 move left and enter state Lr0c.

To check it works we step through the computations for $n = 2 = 10$ and $n = 3 = 11$.

Case of $n = 2$: Start with 1,0 in cells 1 and 2. Machine goes into state IRr and moves to cell 3, then goes back to cell 2 in state IA. It enters state Lr1 and goes back to cell 0, then enters state Ll1 and moves to cell -1.

In cell -1 it writes 1 then enters state Rl and moves to cell 0. It enters state ZRr and moves to cell 1, then enters state Rr and moves right as far as cell 3. It then enters state E and moves to cell 2, writes a blank in cell 2 and moves back to cell 1 in state A. The head moves back to cell 0 in state Lr1, then to cell -1 in state Ll1. This time it steps back to cell -2 and writes a 1 there, then moves to cell -1 in state Rl.

The head moves to cell 0 in state Rl, then to cell 1 in state ZRr. At this point the computation stops.

Case of $n = 3$: Start with 1,1 in cells 1 and 2. Machine goes into state IRr and moves to cell 3, then goes back to cell 2 in state IA. It enters state Lr0C and goes back to cell 0, then enters state Ll0C and moves to cell -1.

In cell -1 it writes 0 then enters state RlC and moves to cell 0. It enters state ZRrC and moves to cell 1, then enters state RrC and moves right as far as cell 3. It then enters state EC and moves to cell 2, writes a blank in cell 2 and moves back to cell 1 in state AC. The head moves back to cell 0 in state Lr0C, then to cell -1 in state Ll0C. This time it steps back to cell -2 and writes a 0 there, then moves to cell -1 in state RlC.

The head moves to cell 0 in state RlC, then to cell 1 in state ZRrC. It moves back to cell 0 in state Fr, then moves back to cell -3 in state Fl, writes 1, and stops.

4. Since I am allowed more latitude this time I assume that the inputs are written in binary to the right of cell 0, the smaller one is padded with zeroes so both arguments are the same length, and they are demarcated by A, B and C. For example a typical input could be A010B100C where the asterisk is in cell 0. As the computation proceeds I will overwrite the inputs as follows: the most recent digit used is overwritten with a D and the rest are overwritten with E. So for eaxmple when I am have used 2 digits of input my exampel above becomes A0DEB1DEC

   States: too many to list!

   Start: on A move right and enter state IR

   IR: move right on all symbols except C, on C move left and enter state IAa.

   IAa: on 0 write D, move left and enter state ILa0. On 1 write D, move left and enter state ILa1.

   ILa0: move left on all symbols except B, on B move left and enter state IAa0.

   ILa1: move left on all symbols except B, on B move left and enter state IAa1.

   IAa0: on 0 write D, move left and enter state ILb0. on 1 write D, move left and enter state ILb1.

   IAa1: on 0 write D, move left and enter state ILb1. on 1 write D, move left and enter state ILb0C.

   ILb0: move left on all symbols except A, on A move left and enter state R0.

   ILb1: move left on all symbols except A, on A move left and enter state R1.

   ILb0C: move left on all symbols except A, on A move left and enter state R0C.

   R0: write 0, move right and enter state R.

   R1: write 1, move right and enter state R.

   R0C: write 1, move right and enter state RC.

   R: on 0 or 1 move right. On A move right and enter state ZR.

   RC: on 0 or 1 move right. On A move right and enter state ZRC.

ZR: on D move left and stop. On 0 or 1 move right and enter state Rb.

ZRC: on D move left and enter state F. On 0 or 1 move right and enter state RbC.

F: on all symbols but blank move left, on blank write 1 and stop.

Rb: on all symbols but B move right, on B move right and enter state Ra.

RbC: on all symbols but B move right, on B move right and enter state RaC.

Ra: on 0 or 1 move right, on D write E, move left and enter state Aa.

RaC: on 0 or 1 move right, on D write E, move left and enter state AaC.

Aa: on 0 write D, move left and enter state La0. On 1 write D, move left and enter state La1.

AaC: on 0 write D, move left and enter state La1. On 1 write D, move left and enter state La0C.

La0: on all symbols but D move left, on D move left and enter state Aa0.

La1: on all symbols but D move left, on D move left and enter state Aa1.

La0C: on all symbols but D move left, on D move left and enter state Aa0C.

Aa0: on 0 move left and enter state Lb0, on 1 move left and enter state Lb1.

Aa1: on 0 move left and enter state Lb1, on 1 move left and enter state Lb0C.

Aa0C: on 0 move left and enter state Lb0C, on 1 move left and enter state Lb1C.

Lb0: move left on all symbols but blank, on blank write 0, move right and enter state R.

Lb1: move left on all symbols but blank, on blank write 1, move right and enter state R.

Lb0C: move left on all symbols but blank, on blank write 0, move right and enter state RC.

Lb1c: move left on all symbols but blank, on blank write 1, move right and enter state RC.

5. If A is not the zero class, $A = a + p\mathbb{Z}$ where p does not divide a. Since p is prime the gcd of p and a is 1 so A has an inverse.

   In what follows I write 1 for the class $1 + p\mathbb{Z}$.

   There is a mistake in the second part. If $p = 2$ there is exactly one class with $A^2 = 1$, namely $A = 1 + 2\mathbb{Z}$.

   For $p > 2$ the answer is as follows. Let $A = a + p\mathbb{Z}$ with $A^2 = 1$. Then $a^2 - 1$ is a multiple of p, and so as p is prime either p divides $a - 1$ or $a + 1$. That is $A = 1 + p\mathbb{Z}$ or $-1 + \mathbb{Z}$.

When $n = 8$ each of the classes $1 + 8\mathbb{Z}$, $3 + 8\mathbb{Z}$, $5 + 8\mathbb{Z}$, $7 + 8\mathbb{Z}$ has square $1 + 8\mathbb{Z}$.

6. Solution 1: by Euler $A^{\phi(n)} = 1$, so $A^{\phi(n)-1}$ is an inverse.

   Solution 2: the list of powers of A must contain a repetition as there are only finitely many classes. Say $A^i = A^j$ with $i < j$. If B is the inverse of A then multiply by B^i and conclude $A^{j-1} = 1 + n\mathbb{Z}$.