CS 1050 Homework 6 Solutions

1.a Proof. We know that $10 \equiv 1\ (mod\ 9)$. That is $10^n \equiv 1\ (mod\ 9)\ \forall n \in \mathbb{N}$. So, we have that $10^n \equiv 10^k\ (mod\ 9)\ \forall n,\ k \in \mathbb{N}$.

b. Proof. Let N be the number in base 10, and $f(N)$ the number obtained by rearranging the digits of N. Let n_i be the number in the i^{th} position in N, and is in the $p(i)^{th}$ position in $f(N)$ (all positions are counted from left starting from 1). So, $N - f(N) = \sum_{i=1}^{m} n_i \cdot (10^{i-1} - 10^{p(i)-1})$ where m is the length of the number. From 1.a we have that $10^{i-1} \equiv 10^{p(i)-1}\ (mod\ 9)$ since $i,\ p(i) > 0$. So $10^{i-1} - 10^{p(i)-1} \equiv 0\ (mod\ 9)$. That is, $10^{i-1} - 10^{p(i)-1} = 9c_i$ for some $c_i \in \mathbb{Z}$. Therefore, $N - f(N) = \sum_{i=1}^{m} n_i \cdot 9c_i = 9 \cdot \sum_{i=1}^{m} (n_i \cdot c_i)$ which is divisible by 9. So, $N - f(N) \equiv 0\ (mod\ 9)$.

c. Proof. We know that if $a \equiv b\ (mod\ n)$ and $c \equiv d\ (mod\ n)$ then $ac \equiv bd\ (mod\ n)$ and $a + c \equiv b + d\ (mod\ n)$. Let the i_{th} digit of a number N be n_i. So $N = \sum_{i=1}^{m} n_i \cdot 10^{i-1}$ where m is the length of the number. Now, $N \equiv \sum_{i=1}^{m} n_i \cdot 10^{i-1}\ (mod\ 9)$. Also, $n_i \cdot 10^{i-1} \equiv n_i\ (mod\ 9)$ since $10_{i-1} \equiv 1\ (mod\ 9)$. So, $N \equiv \sum_{i=1}^{m} n_i\ (mod\ 9)$. So, if N is divisble by 9 then $\sum_{i=1}^{m} n_i \equiv 0\ (mod\ 9)$, which means that the sum of its digits is divisible by 9.

2 When n is prime we cannot say that that $n! + 1$ is prime. The reason is that there may be a number between n and $n! + 1$ that could divide $n! + 1$. We can only *claim* that $n! + 1$ is prime under the assumption that n is the largest prime, since then there cannot exist any other primes between n and $n! + 1$. As an example we can see that we see that 5 is a prime, but $5!+1 = 121$ is not a prime because 11 divides it.

3.a Table 1 shows the stages of the stable marriage algorithm. Each row shows the boy to which a girl is engaged after the corresponding stage of the algorithm.

So, the stable marriage is $(a,\ 2)$, $(b,\ 4)$, $(c,\ 3)$ and $(d,\ 1)$.

b. To get a girl optimal marriage we modify the algorithm so that it is the girls that propose to the boys. Table 2 shows the stages of the modified stable marriage algorithm. Each row shows the girl to which a boy is engaged after the corresponding stage of the modified algorithm.

Stage	1	2	3	4
a proposes to 1	a	none	none	none
b proposes to 2	a	b	none	none
c proposes to 1	c	b	none	none
d proposes to 2	c	d	none	none
a proposes to 2	c	a	none	none
b proposes to 1	b	a	none	none
d proposes to 1	d	a	none	none
c proposes to 3	d	a	c	none
b proposes to 4	d	a	c	b

Table 1: Boy optimal stable marriage.

So the stable marriage is $(a,\ 2)$, $(b,\ 3)$, $(c,\ 4)$ and $(d,\ 1)$. In this algorithm

Stage	a	b	c	d
1 proposes to d	none	none	none	1
2 proposes to a	2	none	none	1
3 proposes to a	2	none	none	1
4 proposes to d	2	none	none	1
3 proposes to b	2	3	none	1
4 proposes to c	2	3	4	1

Table 2: Girl optimal stable marriage.

girls 2 and 3 get better matches than in the boy optimal algorithm.

c. Consider the instance where every boy has the same order of preference, say $pref(boys)$, and every girl has the same order of preference, say $pref(girls)$. Clearly there is a stable marriage in which the i^{th} girl on $pref(boys)$ is married to the i^{th} boy on $pref(girls)$. However, when we run the stable marriage algorithm on this instance, the boy who is married to the i^{th} girl on $pref(boys)$ is rejected $i-1$ times, so i proposals are required from him. Summing this up we get $\frac{n(n+1)}{2} \approx \frac{n^2}{2}$ proposals, where n is the number of boys or girls, so we get $c = \frac{1}{2}$.

d. Suppose that $M * M'$ is not stable. That means that there are two couples in $M * M'$, (m_1, w_1) and (m_2, w_2) (m_i are men and w_i are women) such that m_1 prefers w_2 over w_1 and w_2 prefers m_1 over m_2. Clearly, the way $M * M'$ is constructed, m_1 is married to w_1 in atleast one of the two stable matchings, say M. Suppose in M, w_2 is married to some m'. Since w_2 prefers m_1 over m_2, she prefers m_1 over m' also, because m_2 is someone who is not worse than m' for w_2. Also, since m_1 prefers w_2 over w_1, we get that M is not stable, which is a contradiction, since M is given to be stable. So our initial assumption is wrong. Therefore $M * M'$ is stable.

4. Proof. We can see that in each break the number of pieces of chocolate increases by exacltly one. Initially there was one piece of chocolate (the entire bar). In the end, we need k pieces of chocolate. This will require exacly $k - 1$ breaks as each break increases the number of pieces of chocolate by exacly 1.

5. Proof. Look at the $1 \times 1 \times 1$ cube in the center of the $3 \times 3 \times 3$ rubic cube. It has got six faces, but none of them are exposed. We need to expose all of its faces to separate it out. Also, each cut by an axe can expose only one of its faces. Therefore, atleast 6 cuts are required.

6.a The statement "Every CoC undergrad owns a laptop" is a statement of the form "for all xxxx there exists a yyyy" and can be mathematically stated as "For every CoC undergrad there is a laptop owned by him/her". Negation of the statement is "There is one CoC undergrad who owns no laptop".

b. The statement "There is a CoC undergrad who owns a laptop" is a statement of the form "There exists a xxxx such that there exists a yyyy" and can be mathematically stated as "There exists a CoC undergrad such that there exists a laptop owned by him/her". The statement can be negated to form "There is no CoC undergrad who owns a laptop".