Homework 7 2/25/2015

Huimin He , section 1

February 25, 2015

1. 7.1 Fermat Test
   We have
   $$gcd(a, 561) = 1$$
   so
   $$gcd(a, 3) = 1$$
   $$gcd(a, 11) = 1$$
   $$gcd(a, 17) = 1$$
   Since $3, 11, 17$ are primes, we have the following by fermat's little thoerem,
   $$a^2 \equiv 1 \pmod 3$$
   $$a^{10} \equiv 1 \pmod{11}$$
   $$a^{16} \equiv 1 \pmod{17}$$
   It is easy to verify that $2, 10, 16$ divides 560. So we can write by multiplying the corresponding factor and get
   $$a^{560} \equiv 1 \pmod 3$$
   $$a^{560} \equiv 1 \pmod{11}$$
   $$a^{560} \equiv 1 \pmod{17}$$
   Since $3, 11, 17$ are pairwisely prime, we can write (by Chinese remainder theorem).
   $$a^{560} \equiv 1 \pmod{lcm(3, 11, 17)}$$
   $$a^{560} \equiv 1 \pmod{561}$$
   It is proved that 561 is a Carmichael number.

2. 7.7 Fibonacci numbers:
   (b) d_n be the number of binary digits of Fn.
   $$d_n = \lceil\ log_2 F_n \rceil\ + 1$$
   From part (a) we have $F_n = \frac{1}{\sqrt{5}} r^n$ when n is large. So far large n., we have
   $$d_n \sim log_2 \frac{1}{\sqrt{5}} r^n$$

$$d_n \sim log_2 1/\sqrt{5} + nlog_2 r$$

$$d_n \sim nlog_2 r$$

where r is the Golden Ratio so $a = log_2 r$, $b = 1$,$c = 0$.

(c) Because binary length of n is proportional to its magitude, to compute Fn from the formula (1) or the asymptotic formula in part (a) the exponent is n. so the calculation would need exponential time to get Fn.

3. 7.8

part(c) the graph can be color with chromatic coloring using greedy coloring. There exists permutation such that greedy coloring gives the optimal solution. Consider this permutation that all vertices of color 1 are listest first in any order. Then, vertices with color 2 are listed. Followed by vertices with color 3.. and so on. If greedy coloring is run for such a configuration. Vertices will receive color as in the optimal coloring or a lower valued color. They can not receive a larger color since no two adjacent vertices in optimal coloring share the same color.

part(d) (Implement greedy coloring in linear time) let $l.color$ be the array of the color of all the vertices in order. $l.color$ has size $|V|$. $l.used$ be the array of the color used for coloring. $l.used$ has size $D + 1$ where D is the maximum in degree for all vertices.

Input: adj list representation of graph G.
Output: a legal coloring using less equal than $D + 1$ colorings.

Pseudocode:

```
Initialization
0     set all entires of l.color and l.used to 0
0     for each vertex v in G
Coloring
1     for each vertex v in G
2         for each vertex w in v.adj
3             if l.color(w) > 0
4                 l.used(l.color(w)) = 1
5             l.color(v) = search(l.used)
6         for each vertex s in v.adj
7             if l.color(w) > 0
8                 l.used(l.color(w)) = 0

search(list)
0     for i from 0 to length(list)
1         if list(i) == 0
2             return i
3     end
```

4. 7.9 coloring planar graphs. We can solve this problem using recursion.

   0 find a vertex v_i with degree 5 or less
   1 omit vi from the graph and color the rest the of the graph recursively with 6 colors
   2 color the omitted vertex with the free color.

   For the actual implementation of this, we need to construsct an array that record the color of the vertices. Pass the array as argument to each of recursive call after omitting the vertex that has less than 5 indegrees. Then the call would return the color of the vertices from base case to the upper level call. Then combine the retunred array and the omitted vertex and return the combined array.

Extracredit

1. 7.1 **part(b)** let c be primitive root of p. and let d be primitive root of r. We pick a such that $a \equiv c \pmod{p}$ and $a \equiv dpmodr$. Then we want to show

$$a^{pr-1} \not\equiv 1 \pmod{pr}$$

.

Assume that

$$a^{pr-1} \equiv 1 \pmod{pr}$$

we want to prove by contradition to show this is impossible. If

$$a^{pr-1} \equiv 1 \pmod{pr}$$

, since p, r are distinct primes, then we have

$$a^{pr-1} \equiv 1 \pmod{p}$$

So we have

$$c^{pr-1} \equiv 1 \pmod{p}$$

since c is smaller than p. So

$$c^{pr} \equiv c \pmod{p}$$

Because by fermat's little thoerem

$$c^{p-1} \equiv 1 \pmod{p}$$

We have

$$c^{(p-1)r}c^r \equiv c \pmod{p}$$

We then have

$$c^r \equiv c \pmod{p}$$

so

$$c^{r-1} \equiv 1 \pmod{p}$$

From the property of primitive root we can deduce that $p-1|r-1$ because $c^j \not\equiv 1 \pmod{p}$ for $0 < j < p-1$. We find that $r-1$ is a multiple of $p-1$.
Use the other prime r do the same thing for $a^{pr-1} \equiv 1 \pmod{r}$. Use d the primitive root of r. We can get that $r-1|p-1$. So we have $r-1 = p-1$ and get $p = r$. However, this contradicts with the premise that p, r are disctint primes. By contradiction, pr is not Carmichael number is proved.

2. 7.1 **part(c)** Let the number be n and we want to show how many fermat witness we have. For $a_1, a_2, a_3..., a_i$ that are less than n and $gcd(a_i, n) = 1$. We let

$$a_i^{n-1} \equiv 1 \pmod{n}$$

Also we have $a^{n-1} \not\equiv 1 \pmod{n}$. So $a^{n-1}a_i^{n-1} \not\equiv 1 \pmod{n}$. $a_1, a_2, a_3..., a_i \bmod n$ are numbers smaller than n. the set $(a_1, ..a_i)$ and the set $(a_1a \bmod n, a_2a \bmod n, ..., a_ia \bmod n)$ are of the same size. And they are distinct numbers. So the total size of the two sets must be less than all numbers less than n. So the set $(a_1, ..., a_i)$ cardinality is less than half n. So we have more than $n/2$ fermat witnesses.

3. 7.7 Fibonacci numbers
   The fibonacci matrix is of the form $(f_{n+1}, f_n; f_n, f_{n-1})$. We can use the modulo exponential repeated spuare method to calculate $f_n \bmod m$. Use the same algorithm taught during class. Instead of calculating M^e we calculated A^n where A is the $[0, 1; 1, 1]$ matrix. take mod p each step. In the end take the $(0, 1)$ or $(1, 0)$ entry as answer.

4. 7.8 part (B) greedy coloring is terrble
   For a bipartite graph. let all vertices on the left connect to all the vertices on the right. If we color from vertices $1, 2, 3, 4$ up to n, we will end up using $n/2$ colors. The graph is shown below.(each vertex on the left is connected to all vertices on the right). So every 2 vertices we have to add one more color because the previous vertices on the other side are all connected to the vertex.

5. 6.15XC

Given $N = pq, K = c(p-1)(q-1)$, where c is constant. We want to be able to decyper a text C and get original message M.

We also have the ciphertext $C \equiv M^e \pmod{n}$ (M is the original message) and the public key e. Denote d as the private key we generate to decipher. Let $d = (1-K)/e$.

$$C^d \equiv (M^e)^d \equiv (M^e)^{(1-K)/e} \equiv M^{1-K} \pmod{n}$$

Claim that $M^{1-K} \equiv M \pmod{n}$. Prove:

$$M^{1-K} \equiv M^{1-c(p-1)(q-1)} \pmod{p}$$

$$\equiv M(M^{p-1})^{-c(q-1)} \pmod{p}$$

We have $M^{p-1} \equiv 1 \pmod{p}$ from fermat's little thoerem. Thus

$$M^{1-K} \equiv M^{(1)^{-c(q-1)}} \pmod{p}$$

$$\equiv M \pmod{p}$$

Similarly we can have $M^{1-k} \equiv M \pmod{q}$. By chinese remainder theorem we have and $n = pq$ p, q distinct,

$$M^{k-1} \equiv M \pmod{n}$$

. So the key $d = (1-K)/e$ works to decipher. The algorithm is polynomial time because it only requires some step of modulo exponential calculations. By the method taught during class, we can compute the for $M^{1-K} \bmod p$ in polynomial time.