

# CS 70, Summer 2014 — Homework 4

Harsimran (Sammy) Sidhu, SID 23796591

July 27, 2014

Collaborators: Chonyi Lama, Jenny Pushkarskaya

Sources:

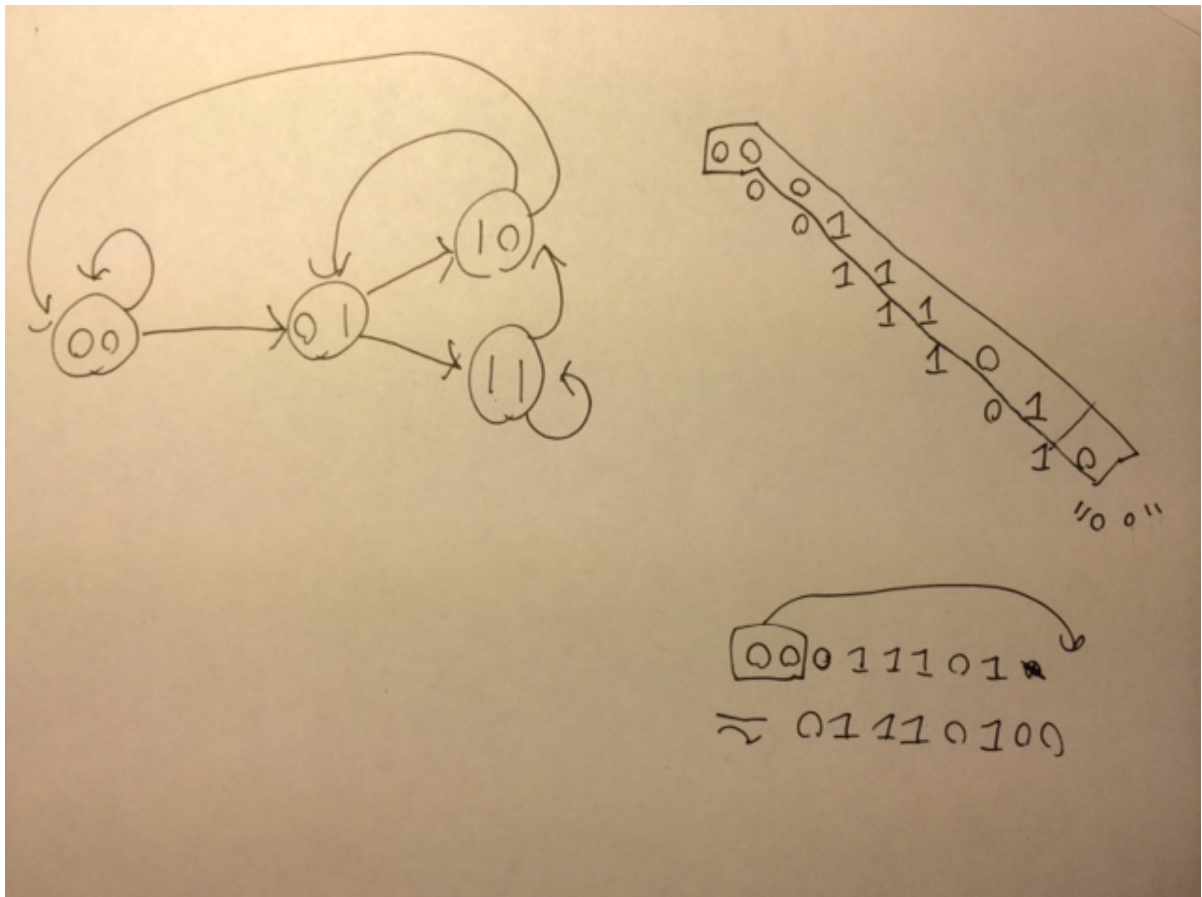
## Problem 1

- (a) Let  $G = (V, E)$  and let every vertex of  $G$  be of even degree. So now  $G$  has zero odd degree vertices which is even. Let us now create an edge between any two vertices. These two vertices now have an odd degree. So now we have two vertices of odd degree which is also even. We can use the fact that an edge connects two vertices so when two even degree vertices are connected with an edge, we now have two more odd degree vertices. So we can only have an even number of odd degree vertices.
- (b) Suppose  $G$  is connected and has  $2c$  odd degree vertices where  $c$  is a positive integer. Let's start at a odd degree vertex  $A$  that has  $2k + 1$  degrees where  $k$  is a natural number. Let's exit vertex  $A$  and each delete edge that we use so we will only travel each edge once. Vertex  $A$  now has a degree of  $2k$  which is even and we are now at some other vertex. We now continue travel to nodes on the graph deleting each edge we use. Vertices with an even degree will have an equal amount of times we enter and exit due to the fact that you have to exit everytime you enter. This will happen until we travel to a vertex that has a degree of 1. When we reach this vertex, the prior edge will be deleted and the degree of this vertex will now be zero and we'll be stuck. This can only happen if the original degree of the vertex was odd since you had to exit for every entrance. From this we can see that if you start from an odd degree vertex you will end at another odd degree vertex trying to cover as many edges with 1 path.

If we follow this pattern of initially starting from an odd degree node and walking until you are stuck we can prove that we can walk on every edge using  $c$  paths. If we start at an odd degree node and walk until we are stuck at some other odd degree node, we eliminate both odd degree nodes and their edges and at least 2 edges from each other node we used in our path. Since each path removes 2 odd degree nodes and  $G$  has exactly  $2c$  odd degree nodes we can say that we only need  $c$  paths to remove every edge.

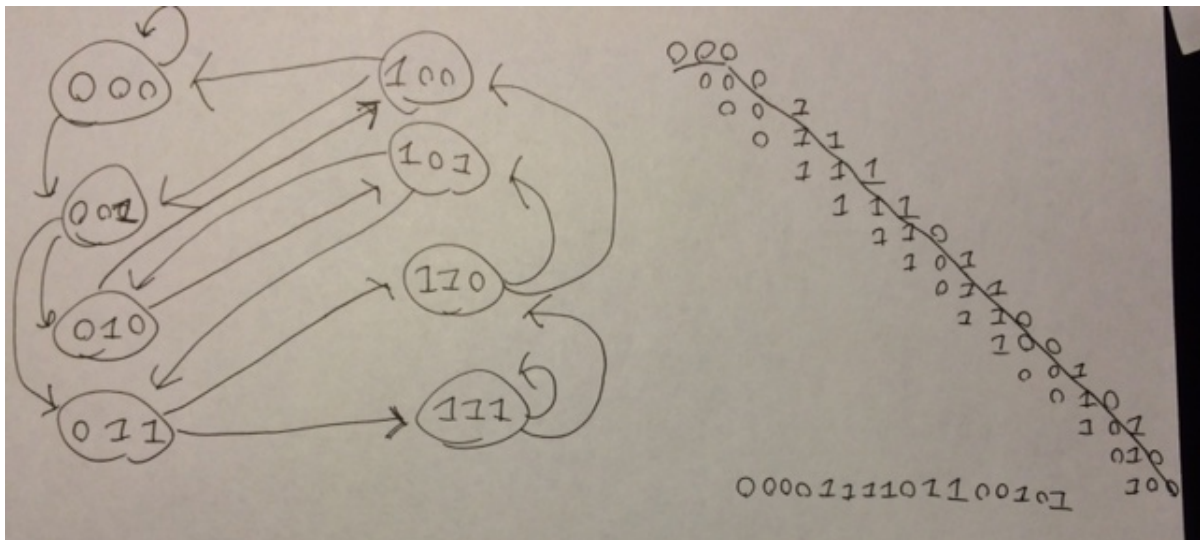
## Problem 2

- (a) It makes sense that we can use a directed graph for this since it is sort of like taking a path where you can only go to a node if the end of your current node has an overlap with the start of the next node. For this graph to be useful for us we construct with our nodes being 2-bit tuples and connect with directed edges that overlap of 1-bit from the end of one node to the start of another one. This makes sense since a 3-bit tuple is simply the overlap of two 2-bit ones. After creating the 4 2-bit nodes we connect them with directed edges covering every possible 1-bit overlap including self-loops. To construct the sequence, we simply have to perform an Eulerian Tour from any node. We know we can perform a tour due to the fact that every node has an equal amount of entrance and exit edges and the total amount of edges for each node is even. I choose the 00 node to start.



- (b) Using the method from part a, I chose to use 3-bit tuples as nodes and directed edges from overlaps of the end of one node to the start of another using an overlap of 2-bits. I then performed the same task of conducting an Eulerian Tour on the directed graph.

My final sequence was 0000111101100101 and I confirmed it contains all 4-bit sequences only once.



### Problem 3

(a)  $\binom{p}{k} = \frac{p!}{(p-k)!k!} \pmod{p}$  for  $0 < k < p$

$$\binom{p}{k} = \frac{p \times (p-1)!}{(p-k)!k!} \pmod{p} \text{ for } 0 < k < p$$

$$\binom{p}{k} = \frac{0 \times (p-1)!}{(p-k)!k!} \equiv 0 \pmod{p} \text{ for } 0 < k < p$$

$$\binom{p}{k} = 0 \pmod{p} \text{ for } 0 < k < p$$

(b) show  $(x+p)^p = x^p + y^p \pmod{p}$

$$(x+y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k, \text{ where } \binom{n}{k} = \frac{n!}{(n-k)!k!}$$

$$(x+p)^p = \sum_{k=0}^p \binom{p}{k} x^{p-k} y^k \pmod{p}$$

Since  $\binom{p}{k} = 0 \pmod{p}$  for  $0 < k < p$  from part a

$$(x+p)^p = \sum_{k=0}^p \binom{p}{k} x^{p-k} y^k = \binom{p}{0} x^{(p-0)} y^0 + \binom{p}{p} x^{(p-p)} y^p \pmod{p}$$

$$(x+p)^p = \binom{p}{0} x^p + \binom{p}{p} y^p \pmod{p}$$

$$(x+p)^p = \frac{p!}{(p-0)!0!} x^p + \frac{p!}{(p-p)!p!} y^p \pmod{p}$$

$$(x+p)^p = x^p + y^p \pmod{p}$$

(c) Proof:  $a^p = a \pmod{p}$  by Strong Induction

Base Cases:

let  $a = 0$

$$0^p = 0$$

let  $a = 1$

$$1^p = 1$$

Inductive Hypothesis: Assume for some  $x \in \mathbb{N}$  ( $0 \leq x < a$ )  $x^p = x \pmod{p}$

Using  $(x+y)^p = x^p + y^p \pmod{p}$  from part b

$$a^p = ((a-1) + 1)^p = (a-1)^p + 1 \pmod{p}$$

$$a^p = (a-1)^p + 1 \pmod{p}$$

Now let  $a = x + 1$  for the inductive step

$$(x+1)^p = (x+1-1)^p + 1 = x^p + 1 \pmod{p}$$

$$(x+1)^p = x + 1 \pmod{p}$$

Substitute Inductive Hypothesis

□

## Problem 4

- (a) There are  $13!$  ways to form the 13-bit string. Since 5 of the bits are 1 we can divide by  $5!$  and since 8 of the bits are 0 we can also divide by  $8!$ . so

$$\text{ways} = \frac{13!}{5!8!}$$

- (b) There are  $2^{55}$  ways to make a 55 bit string due to the fact that each bit has 2 choices and there are 55 bits in total. Knowing this, Half of these contain more 0 than 1's. So the amount of 55 bit strings with more 1's than 0 is

$$\text{ways} = \frac{2^{55}}{2} = 2^{54}$$

- (c) There are 52 cards in a deck and we choose 13 of them so the answer would be 52 choose 13.

$$\binom{52}{13} = \frac{52!}{(52-13)!13!} = \frac{52!}{39!13!}$$

- (d) There are 52 cards in a deck and 4 of them are aces so if we ignore them we only have 48 to choose the 13 we need.

$$\binom{48}{13} = \frac{48!}{(48-13)!13!} = \frac{48!}{35!13!}$$

- (e) There are 4 aces and we choose all 4 of them so there is only 1 way to choose the aces. We now have 9 cards to choose from a deck of 48.

$$\binom{48}{9} = \frac{48!}{(52-9)!9!} = \frac{48!}{(43!9!)}$$

- (f) We first choose 5 spades from the 13 in the deck and then multiple the amount of ways to get the 5 spades from 13 and then remove the spades from the deck. We then find the amount of ways to fill the rest of our hand with the remaining deck.

$$\binom{13}{5} = \frac{13!}{8!5!}$$

$$\binom{39}{8} = \frac{39!}{(39-8)!8!} = \frac{39!}{31!8!}$$

$$\text{ways} = \binom{13}{5} \times \binom{39}{8} = \frac{39!13!}{31!8!8!5!}$$

- (g) There are 52 ways to pull a card out of a full deck. After pulling out 1 card, there are now 51 ways to pull out a card, and then 50 and so forth.

So the amount of ways to order a deck is  $52!$

- (h) "KENTUCKY" has 8 letters, one of which is repeated once. So the amount of ways to order the word is  $8!$  due to the fact that when you pull out a letter you have  $n - 1$  choices. Knowing

the amount of ways to order the words and the fact that we have a repeated letter. The amount of anagrams we have is  $\frac{8!}{2!}$  since it doesn't matter which "K" is in what place.

- (i) We can order "ALASKA"  $6!$  ways due to the fact that it has 6 letters. However "A" is repeated twice so have to divide the total order by  $3!$ . So the amount of anagrams is  $\frac{6!}{3!}$
- (j) "MISSISSIPPI" has a total of 11 letters. This means we can order the word  $11!$  ways. However the letter has 4 S's, 4 P's and 2 I's. so the amount of anagrams we can have is  $\frac{11!}{4!4!2!}$  Since we don't worry about order.