Introduction to Computer Science
HW #3
Due: 2015/04/29

Chapters 4, 5 Review Problems (Ch4: 8% each, Ch5: 13% each)**:**

Ch 4.

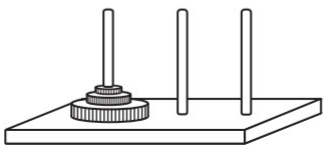
***39.** In a network based on the bus topology, the bus is a nonshareable resource for which the machines must compete in order to transmit messages. How is deadlock (see the optional Section 3.4) controlled in this context?

***41.** Why does the transport layer chop large messages into small packets?

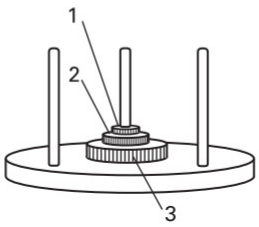
***45.** At what layer in the TCP/IP protocol hierarchy could a firewall be placed to filter incoming traffic by means of
a. Message content
b. Source address
c. Type of application

Ch 5.

40. The puzzle called the Towers of Hanoi consists of three pegs, one of which contains several rings stacked in order of descending diameter from bottom to top. The problem is to move the stack of rings to another peg. You are allowed to move only one ring at a time, and at no time is a ring to be placed on top of a smaller one. Observe that if the puzzle involved only one ring, it would be extremely easy. Moreover, when faced with the problem of moving several rings, if you could move all but the largest ring to another peg, the largest ring could then be placed on the third peg, and then the problem would be to move the remaining rings on top of it. Using this observation, develop a recursive algorithm for solving the Towers of Hanoi puzzle for an arbitrary number of rings.



41. Another approach to solving the Towers of Hanoi puzzle (Problem 40) is to imagine the pegs arranged on a circular stand with a peg mounted at each of the positions of 4, 8, and 12 o'clock. The rings, which begin on one of the pegs, are numbered 1, 2, 3, and so on, starting with the smallest ring being 1. Odd-numbered rings, when on top of a stack, are allowed to move clockwise to the next peg; likewise, even-numbered rings are allowed to move counterclockwise (as long as that move does not place a ring on a smaller one). Under this restriction, always move the largest-numbered ring that can be moved. Based on this observation, develop a nonrecursive algorithm for solving the Towers of Hanoi puzzle.



Introduction to Computer Science
HW #3
Due: 2015/04/29

Programming Problem (50%):

First, VERY IMPORTANT: check if sizeof(unsigned long long int) or sizeof(unsigned long int) is 8. If not, use another computer.

Write two pieces of code:

(a) cipher.cpp reads the file "plain.txt" containing one string (length < 10000) and "public_key.txt" containing *N* and *e*. cipher.cpp should then output "secret.txt" as integers encrypted by RSA. The encoding concatenates 2 chars into one integer. For example, "AB" would be encoded as $(65*2^8+66)=16,706$. If only one char remains, put it to leftmost. For example, "A" would be encoded as $65*2^8=16,640$.

(b) decipher.cpp reads the file "secret.txt" and "private_key.txt" containing *N* and *d*. decipher.cpp should then output "message.txt" with content same as "plain.txt".

Note: Be careful about overflow, signed/unsigned, and eof() problem. "Ned.txt" contains more (*N*,*e*,*d*) sets for you to test.

Bonus (5%)

Write the following function:

unsigned long long int findD(**unsigned long long int** e, **unsigned long long int** phi)

, which returns *d*, where $de \equiv 1 \pmod{phi}$. Save the function into bonus.cpp. No main(). Note: You need to use Euclidian algorithm. Enumeration won't earn any credit.