EE 607 Spring 2017
Homework 1 Total Points = 6

Hualiang Li

- **Problem 1** [1 pt.]: Use the program *traceroute* on wiliki (or from a Linux virtual machine, e.g., Ubuntu, on your laptop) to find the IP address (in decimal-dot notation) of the following
  - www.staradvertiser.com
  - www.nsf.gov
  - wiliki.eng.hawaii.edu

  and another server of your choosing. The IP address is in the first line of the results from traceroute (in parentheses). traceroute will try to print out the path taken to the server but the server may withhold replies for security reasons. So you may receive asterisks "* * *". At that point, kill traceroute by typing control-c.

  Also give the class of the IP address (A, B, or C).

  - www.staradvertiser.com
    IP: 161.47.5.94
    161 in binary: 10100001, so it is class B.
  - www.nsf.gov
    IP: 128.150.4.107
    128 in binary: 10000000, so it is class B.
  - wiliki.eng.hawaii.edu
    IP: 127.0.0.1
    127 in binary: 11111111, so it is class E.

    google.com
    IP: 216.58.193.110
    216 in binary: 11011000, so it is class C.

- **Problem 2** [1 pt]: Go to the IETF home page at www.ietf.org.
  - Read "About the IETF"
  - Note that there are Tutorials on different networking subjects
  - Go to the RFC Pages --> Search.
    - Search for RTP

  What is the RFC number of RTP: A Transport Protocol for Real-Time Applications (the most updated version dated July 2003)?

  RFC 3550

- **Problem 3** [1 pt]: (SLIP synchronization) Consider the following byte stream that is transmitted by the SLIP protocol. The byte stream is in hexadecimal. The stream contains one complete packet. Write down the bytes of the packet (getting rid of any ESC SLIP bytes).

.... 3f, fe, db, dc, 33, c0, 02, 90, ff, ff, f3, db, dd, 00, ff, fe, f3, c0, f3, 00, db, dc, ...

02, 90, ff, ff, f3, db, 00, ff, fe, f3

The following are problems in the appendices of textbook CLRS:

Problem A.1-1 (page 1149) (1pt)

= (2*1-1) + (2*2-1)+(2*3-1)+(2*4-1) + … + (2*n-1)
= (2*1) + (2*2) + (2*3) + (2*4) + … + (2*n) – n
= 2*(1+2+3+4+…+n)-n
=2*(1+n)*n/2 – n
= n^2

Problem B.4-1 (page 1172) (1pt)

Proof by induction.
Base step: When the graph has only 1 edge, total of degree = 2 = 2*1.
Hypothesis: Suppose that if for n edges, the total of degree is 2*n.
Induction step: then for n+1 edges, the total of degree is 2*(n+1).
Total of (n+1) degree = total of n degree + 2 = 2*n+2 = 2(n+1)
Therefore, we proved that the total of degree is 2* number of edges.

Problem B.5-3 (page 1180) (1 pt) *[Recall that a rooted binary tree is comprised of nodes that are either leaves or internal nodes. A leaf node has no children. An internal node has either one or two children. A node's degree is the number of children it has.]*

Note that there are two parts to this problem which are contained in the first and second sentences. The second part is easy since the definition of a full tree is that it does not have any degree-1 nodes.

Be sure that you identify the base step, induction hypothesis and induction step.
First part:
Base step: If the binary tree has only 1 node, then it consists of a single leaf, no degree-2 nodes, so the number of degree-2 nodes is 1 fewer than the number of leaves.
Induction hypothesis: Suppose that if a binary tree with n nodes, then the number of degree-2 nodes is 1 fewer than the number of leaves.
Induction step: if a binary tree with n+1 nodes, the number of degree-2 nodes is 1 fewer than the number of leaves. There are two cases: if the addition node has a sibling, or no sibling. Because the additional node is not root, it has a parent P,
for the case it has a sibling: P is degree-2 node, so it is degree 1 before adding additional node, so by the induction hypotheses,
number of leaves after adding the addition node = number of leaves before adding the addition node + 1 = number of degree-2 before adding the node + 1 + 1 = number of degree-2 after adding the node+1
for the case it has no sibling, P is degree-1, so it is a leaf, so number of leaves after adding node = number of leaves before adding node and number of degree-2 before adding node = number of degree-2 after adding node, so, the number of degree-2 nodes is 1 fewer than the number of leaves.
Therefore, the number of degree-2 nodes is 1 fewer than the number of leaves.

Second part:
If it is a full binary tree, all the internal nodes are degree-2 nodes, so the number of internal nodes in a full binary tree is 1 fewer than the number of leaves.

Problem B.5-4 (page 1180) (1 pt) You can use any proof method. You are not restricted to prove by mathematical induction. As a suggestion, try proof by contradiction.

Suppose there is a n nodes tree has a height of lower_bound of (lgn) – 1, then because it is a binary tree, the maximum number of node at depth n is 2^n, so the total maximum number of nodes at height of lower_bound of (lgn)-1 is

$1+2+4+\ldots+2^{lower\_bound(lgn)-1} = 2^{lower\_bound(lgn)-1} <= n-1 < n$, therefore, it can never has n nodes, this is contradict to the assumption. Therefore, A nonempty binary tree with n nodes has height at least lower bound of lgn.