CSCI 3104 (Spring 2016): Quiz-3. **Time Limit**: 25 minutes. **Maximum points**: 20.

Your Name: ____________________________________

P1 (2 points) A red black tree T has n elements that have been inserted in some order.
(A) What is the worst case complexity of inserting a new element?
$\boxed{\Theta(\log_2(n))}$

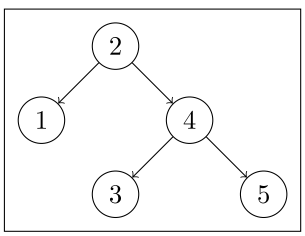
(B) What is the worst case complexity to find if a given key exists in the red-black tree? $\boxed{\Theta(\log_2(n))}$

Suppose we now insert the elements into a hashtable H instead, where the hashtable has $K > n$ slots.
(C) What is the worst case complexity of inserting a new element into H?
$\boxed{\Theta(n)}$

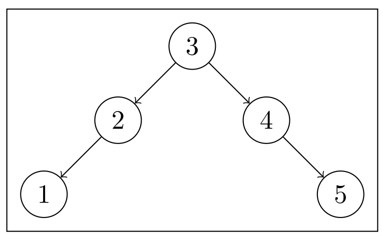
(D) What is the worst case complexity to find if a given key exists in the hashtable H? $\boxed{\Theta(n)}$

(E) What is the average case complexity to find if a given key exists in the hashtable H? $\boxed{\Theta(1) \text{ or } \Theta(\frac{K}{N})}$

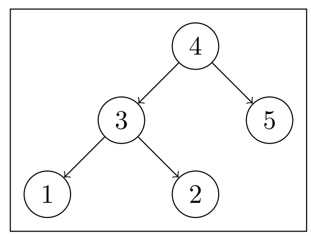
P2 (10 points) Suppose the elements $1, \ldots, 5$ are inserted into a redblack tree in some order (not specified), which of the following red-black trees are legally possible? Fill out the table below and specify the nodes that should be colored red if you think the tree is valid.



(A)



(B)



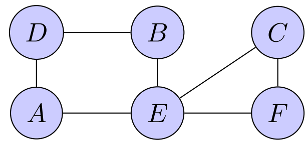
(C)

Part	Legal Red-Black Tree?	Red Nodes
(A)	Yes	4 (or alt. $3, 5$)
(B)	Yes	$1, 5$
(C)	No (not a BST, actually!)	3 (or $1, 2$).

For this problem part (C), we also accepted a "YES" answer.

P3 (5 points) Consider a graph below whose nodes denote people and edges denote friendships on facebook.

We are asked to choose an *independent set* S of nodes (people) such that no two nodes in S are friends with each other (connected by an edge).



(A) Find the largest independent S for the graph above?

The largest independent set is either $\{A, B, C\}$ or $\{A, B, F\}$ of size 3.

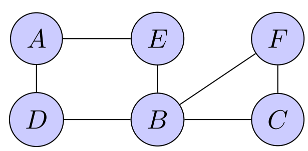
(B) Consider a greedy algorithm for independent as follows:

1. Repeat until the graph is empty:
2. choose the person p with the *least number of friends*.
3. If there are many nodes that are tied for the least number of friends,
4. choose the node p which is alphabetically earliest.
5. add p to the set S
6. remove p and all its friends from the graph.

What the set S provided by the greedy algorithm on the graph above?

The greedy algorithm finds the set $\{A, B, C\}$

(C) Consider the same graph but the names of nodes changed slightly.



(D) What is the set provided by the greedy algorithm on the new graph above?

The greedy algorithm finds $\{A, B\}$

(E) Yes/No: Is the greedy strategy shown above optimal? No