1. (10 pts) Demonstrate the insertion of keys 5, 28, 19, 15, 20, 33, 12, 17, 10 into a hash table with collisions resolved by chaining. Let the hash table have 7 slots, and let the hash function be \( h(k) = k \mod 7 \).

Solution:
2. (8 pts) Consider the following potential red-black trees (with the keys left out). Recall that we represent black nodes by hashed lines and red nodes by no hashed lines. Which of the following are valid red-black trees? For each which is not a valid red-black tree, give a reason why it is not a red-black tree.
Solution: Only II. and IV. are red-black trees.

I. is not a red-black tree because the path from the root to the nil[T] by going right twice goes through only 1 black non-nil node, but the other paths (for example going left 3 times) go through 2 black non-nil nodes.

III. is not a red-black tree because the root is not black.

V. is not a red-black tree since there is a red node with a red child.

3. (6 pts) Consider the following red-black tree $T$. What does the tree look like upon performing $\text{RB-Delete}(T, z)$? You must show intermediate steps.

Solution: Since $z$ is a leaf (except for the NIL’s under it), the initial deletion is straightforward. Then proceed
4. (6 pts) Consider the following red-black tree $T$. What does the tree look like upon performing RB-Insert($T, z$) for $z$ a new node with key value 12? You must show intermediate steps.
5. (3 pts) Suppose we have numbers between 1 and 1000 in a binary search tree and we want to search for the number 502. Can the following sequence be a sequence of nodes examined? Why or why not?


**Solution:** This is a valid sequence of nodes, because, as we can see below, the tree formed by going from the root 723 down to 502 is a valid binary search tree.
6. (10 pts) Consider the following binary search tree $T$.

(a) What is Tree-Successor($T, c$)?

Solution: i.

(b) What is Tree-Successor($T, h$)?

Solution: b.

(c) Put the keys 1,2,3,4,5,6,7,8,9,10 into the nodes of $T$ so that $T$ remains a binary search tree. (In other words, put these numbers into $T$ in the order of an inorder tree walk.)

Solution:
(d) Write down an order of inserting, using Tree-Insert, the numbers 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 into an initially empty binary search tree so that the result is the binary search tree from the previous part (c).

Solution: The idea is to work from the root down, since the root is inserted first into an initially empty binary search tree, and then it cannot move from the root (since the subsequent nodes are all inserted as leaves). So, for example, the sequence 6, 4, 7, 2, 5, 9, 1, 3, 8, 10 works.

7. (10 pts)

(a) If \( h \) is the hash function from the positive integers to \( \{0, 1, 2, 3, \ldots, 99\} \) (so \( m = 100 \)), using the multiplication method for constant \( A = .7329 \), compute \( h(423) \) and \( h(13) \). Show your work.

Solution: \( h(423) = \lfloor m((423 \cdot A) \mod 1) \rfloor = \lfloor 100(310.0167 \mod 1) \rfloor = \lfloor 100(.0167) \rfloor = 1 \).

\( h(13) = \lfloor m((13 \cdot A) \mod 1) \rfloor = \lfloor 100(9.5277 \mod 1) \rfloor = \lfloor 100(.5277) \rfloor = \lfloor 52.77 \rfloor = 52. \)

(b) Let \( h': U \rightarrow \{0, 1, 2, \ldots, 306\} \) (so \( m = 307 \)) be an auxiliary hash function defined by \( h'(k) = k \mod 307 \). Let \( h(k, i) \) be the hash function with \( m = 307 \) made from \( h' \) using linear probing. Compute \( h(5030, 0) \) and \( h(306, 15) \). Show your work.

Solution: \( h(k, i) = (h'(k) + i) \mod m \). So \( h(5030, 0) = h'(5030) = 5030 \mod 307 = 118 \). Note that 5030 \mod 307 can be computed on a calculator as \( \lfloor (5030/307) \mod 1 \rfloor \times 307 \).

\( h(306, 15) = (h'(306) + 15) \mod 307 = (306 + 15) \mod 307 = 321 \mod 307 = 14. \)

(c) What is primary clustering and why does linear probing tend to suffer from it?

Solution: In linear probing, collisions in the hash table are resolved by moving to the next available slot in order. So if all the slots 7 through 18 are filled, then the item will be put in slot 19 for any auxiliary hash value \( h'(k) \) between 7 and 18. So there is a large chance that the cluster from 7 to 18 will grow by taking up the value 19. This phenomenon is called primary clustering. Primary clustering is problematic, since if the clusters are large, then it tends to take a long time to find an open space in the hash table.