Lecture 9-red-black trees

1. What type of data structure is a Red-Black Tree?
   a) Array
   b) Linked List
   c) Balanced Binary Search Tree
   d) Hash Table

   Answer: c) Balanced Binary Search Tree

2. In a Binary Search Tree, where are items smaller than a given node placed?
   a) To the left
   b) To the right
   c) Above
   d) Below

   Answer: a) To the left

3. What is the guaranteed height of a balanced search tree with n items?
   a) O(n)
   b) O(log n)
   c) O(n^2)
   d) O(1)

   Answer: b) O(log n)

4. How many colors can a node in a Red-Black Tree have?
   a) 1
   b) 2
   c) 3
   d) 4

   Answer: b) 2

5. What color are the root and leaves (NIL) in a Red-Black Tree?
   a) Red
   b) Black
   c) Either red or black
   d) No color

   Answer: b) Black

6. If a node is red in a Red-Black Tree, what color must its children be?
   a) Red
   b) Black

c) Either red or black
d) No color

Answer: b) Black

7. In a Red-Black Tree, all paths from a node to its NIL descendants contain:
a) The same number of red nodes
b) The same number of black nodes
c) An equal number of red and black nodes
d) No specific pattern

Answer: b) The same number of black nodes

8. What is the maximum ratio between the longest and shortest path in a Red-Black Tree?
a) 1:1
b) 2:1
c) 3:1
d) 4:1

Answer: b) 2:1

9. What is the time complexity of search, insert, and remove operations in a Red-Black Tree?
a) O(1)
b) O(n)
c) O(log n)
d) O(n^2)

Answer: c) O(log n)

10. What technique is used to fix violations after inserting or removing nodes in a Red-Black Tree?
a) Sorting
b) Rotations
c) Splitting
d) Merging

Answer: b) Rotations

11. What is the primary goal of rotations in a Red-Black Tree?
a) To increase the height of the tree
b) To decrease the height of the tree
c) To change the color of nodes
d) To remove nodes

Answer: b) To decrease the height of the tree

12. What is the time complexity of a rotation operation?
a) O(1)
b) O(log n)
c) O(n)
d) O(n^2)

Answer: a) O(1)

13. How many main scenarios are there after inserting a node Z in a Red-Black Tree?
a) 2
b) 3
c) 4
d) 5

Answer: c) 4

14. In Case 0 of insertion, what action is taken when Z is the root?
a) Color Z red
b) Color Z black
c) Rotate Z
d) No action needed

Answer: b) Color Z black

15. In Case 1 of insertion, what is done when Z's uncle is red?
a) Rotate Z
b) Recolor Z's parents and grandparent
c) Remove Z
d) No action needed

Answer: b) Recolor Z's parents and grandparent

16. In Case 2 of insertion (triangle case), what action is taken?
a) Rotate Z
b) Rotate Z's parent
c) Rotate Z's grandparent
d) No rotation needed

Answer: b) Rotate Z's parent

17. In Case 3 of insertion (line case), what actions are taken?
a) Rotate Z's grandparent only
b) Recolor Z's parents and grandparent only
c) Rotate Z's grandparent and recolor Z's parents and grandparent

d) No action needed

Answer: c) Rotate Z's grandparent and recolor Z's parents and grandparent

18. What is the overall time complexity of inserting a node in a Red-Black Tree?
a) O(1)
b) O(log n)
c) O(n)
d) O(n^2)

Answer: b) O(log n)

19. Which of the following is NOT an application of Red-Black Trees?
a) Java's TreeMap
b) C++ STL's map
c) Linux kernel's completely fair scheduler
d) Python's list implementation

Answer: d) Python's list implementation

20. What property of Red-Black Trees makes them efficient for use as system symbol tables?
a) They use only two colors
b) They guarantee O(log n) time complexity for basic operations
c) They always have a black root
d) They require frequent rotations

Answer: b) They guarantee O(log n) time complexity for basic operations