110612117 張仲瑜 演算法概論 Exercise#2 Report

1. **Environment**

   OS: windows 10, compile version: g++ 14, IDE: VSCode

   Compile and Run in VScode Terminal:

   g++ Exercise2_110612117.cpp -o Exercise2_110612117 ; .\Exercise2_110612117.exe

2. **Results**

   a. **Method (Core)**

      i. **Insertion**

         1. If empty tree → create black tree root → done
         2. Traverse → insert new leaf node as read
         3. If parent is black → done
         4. If uncle is not NULL and is red → push the two red to grandparent,
            parent, uncle → black, grandparent → red (if not root)
            newNode → grandparent to recursively check red-red conflict.
         5. else → LL shape → right rotate, RR shape → left rotate,
            parent → black, grandparent → red,
            RL shape→ left rotate on parent than right rotate on grandparent,
            LR shape → right rotate on parent than left rotate on grandparent.
            parent → black, grandparent → red

      ii. **Deletion**

         1. Find successor first, successor = itself when not found.
         2. Delete the note and replace it with successor, than delete the successor
         3. If successor is red with no children→ delete it since no rule is affected. → done
         4. If successor is black with only one red child → delete successor, replace with its child and change child’s color to black → done
         5. Otherwise 6 cases need to be recursively handled until cases 1,4,6 is reached.
            case 1: the double black node is root → done
            case 2: the sibling of the double black node is red
            L rotate on parent, sibling → black, parent → red
            case 3: black parent, black sibling with 2 black child
            sibling → red, successor → parent

case 4: red parent, black sibling with 2 black child

parent → black, sibling → red → done

case 5: black sibling with 1 left red and 1 right black children

right rotate on sibling, sibling → red, sibling' s red child → black

(this is the case when sibling is on the left hand side, the mirror one is similar)

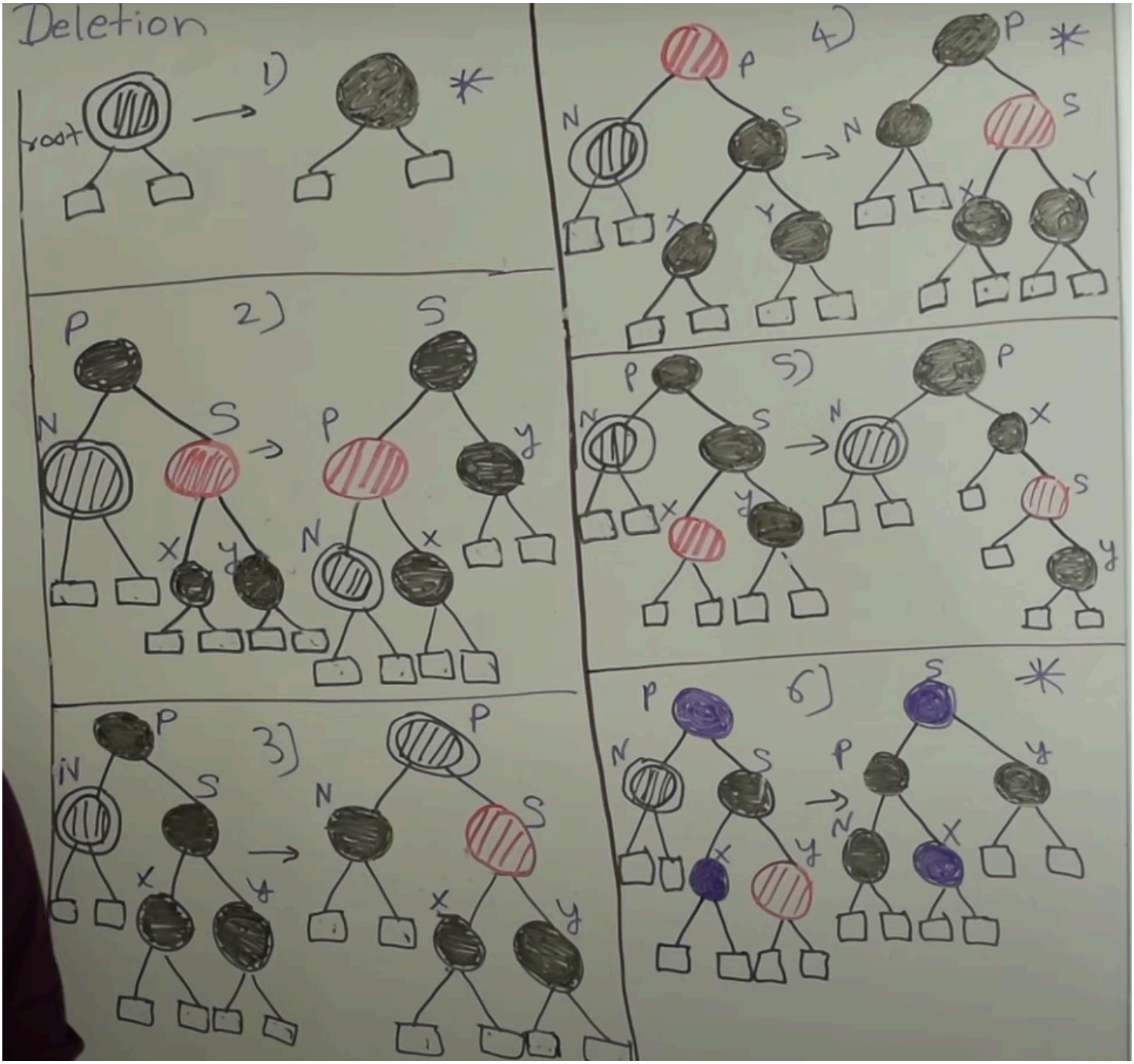
case 6: black sibling with 1 red right child

left rotate on parent and sibling' s red child → black → done

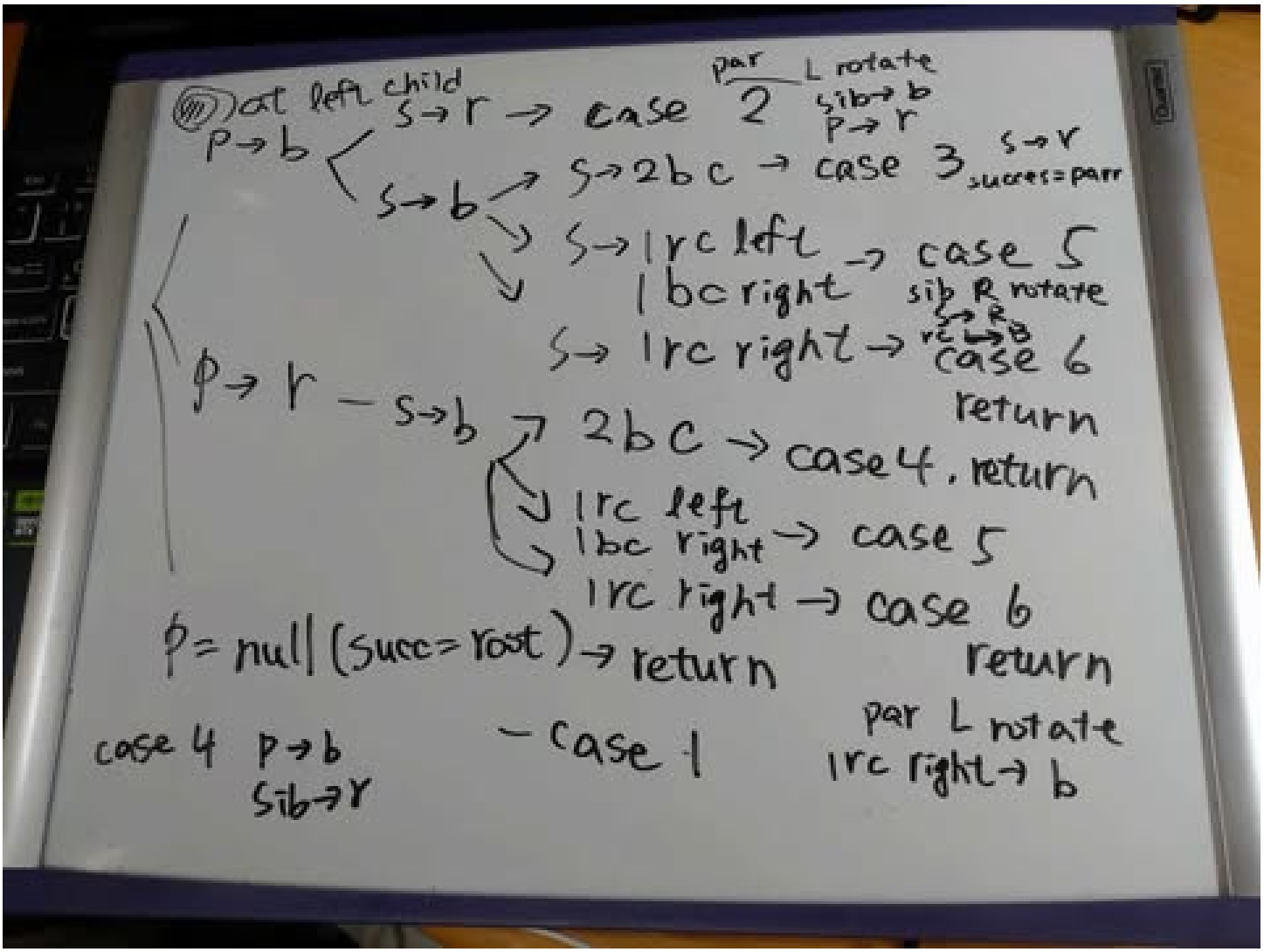
b. **Running time**

   i. **Since the red-red conflict restricts the maximum height of the tree to be less than twice the length of the shortest path, the time complexity of insertion, traverse, and delete are all O(logn), n = the number of nodes.**

c. **Further discussion**



- **The graph of 6 cases drawn by [Tushar Roy]**



- **My handscript**