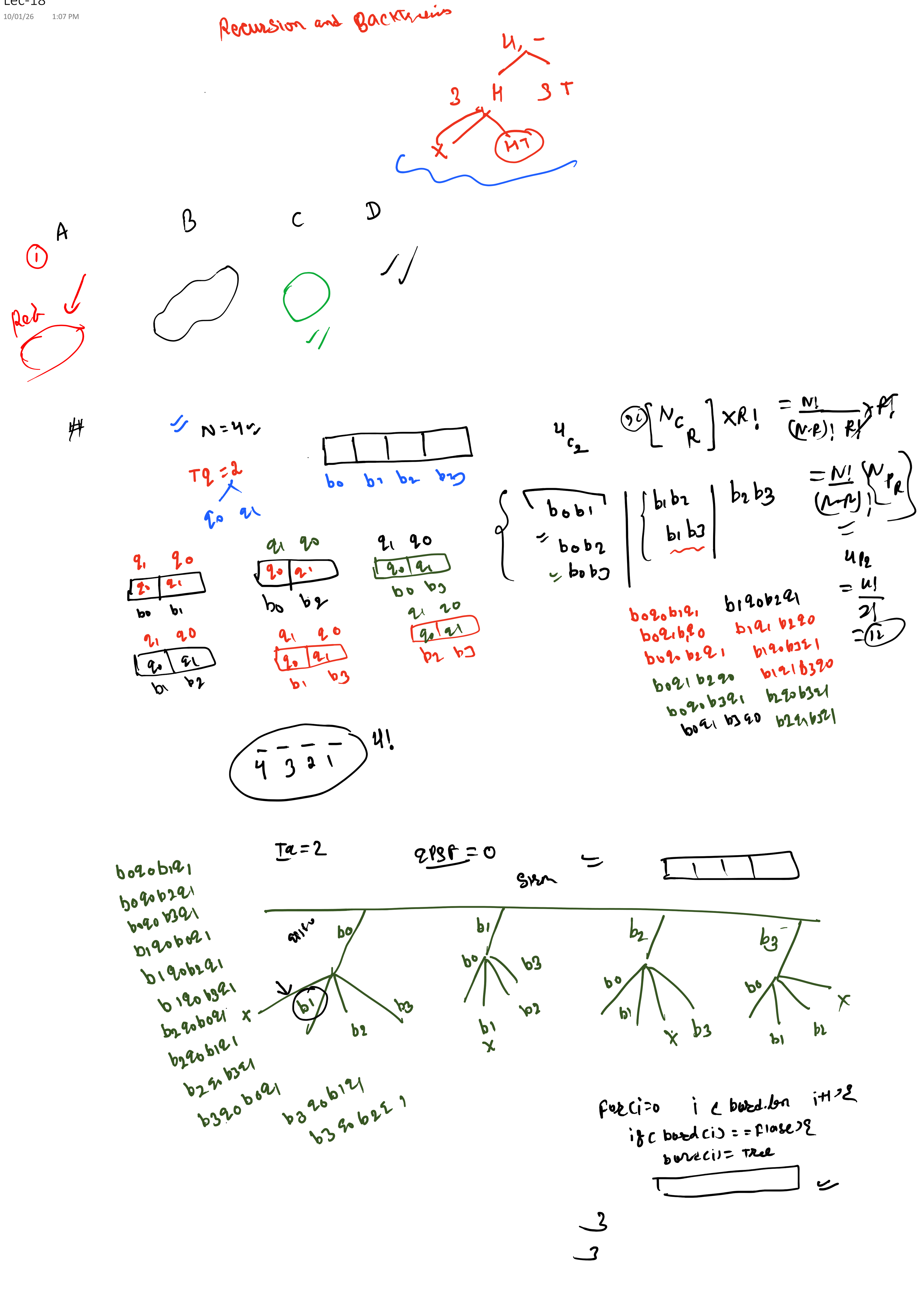
Lec-18

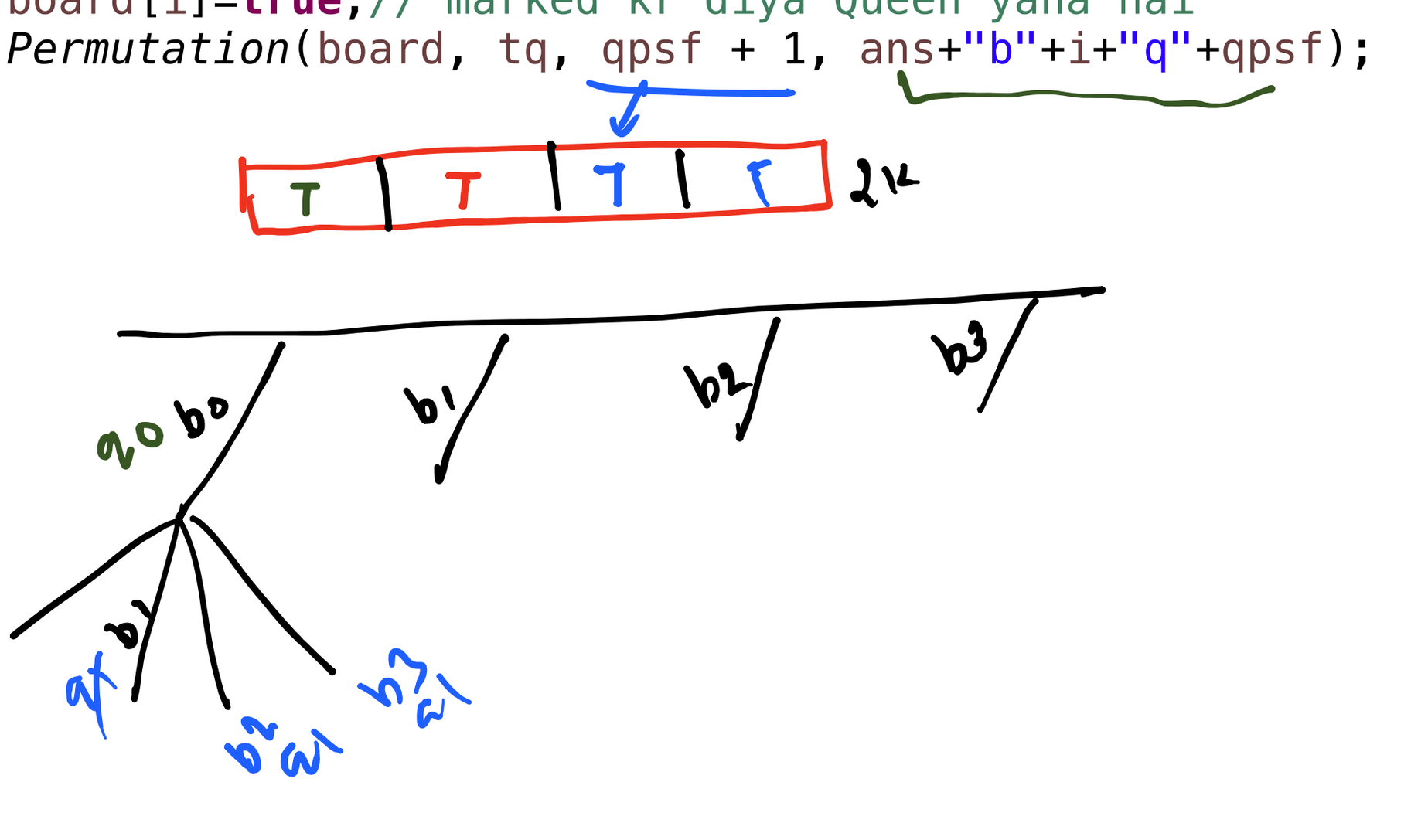
10/01/24 1:07 PM

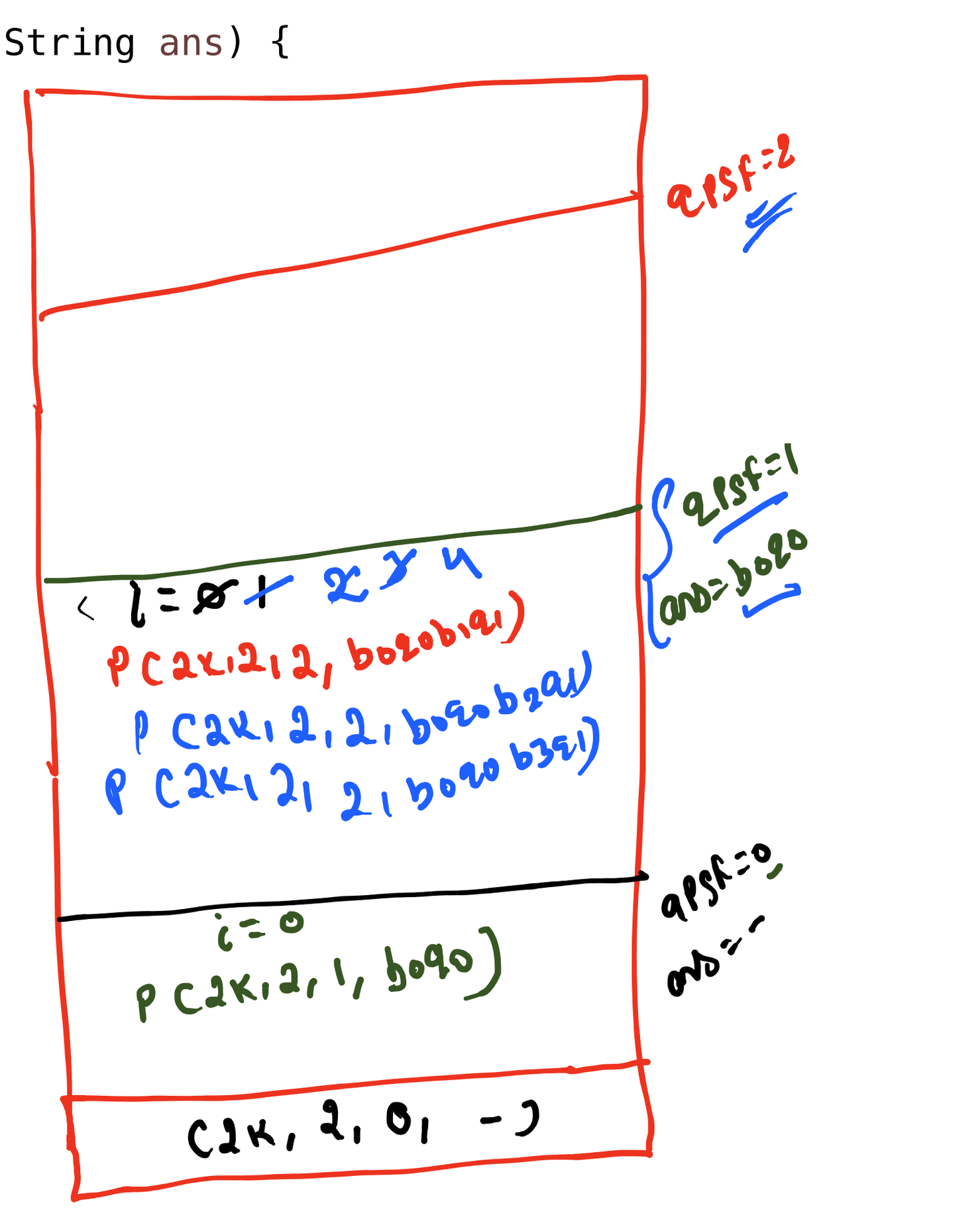
Recursion and Backtracking



```java
public static void Permutation(boolean[] board, int tq, int qpsf, String ans) {
    if (tq == qpsf) {
        System.out.println(ans);
        return;
    }

    for (int i = 0; i < board.length; i++) {
        if (board[i] == false) {
            board[i]=true;// marked kr diya Queen yaha hai
            Permutation(board, tq, qpsf + 1, ans+"b"+i+"q"+qpsf);
        }
    }
}
```

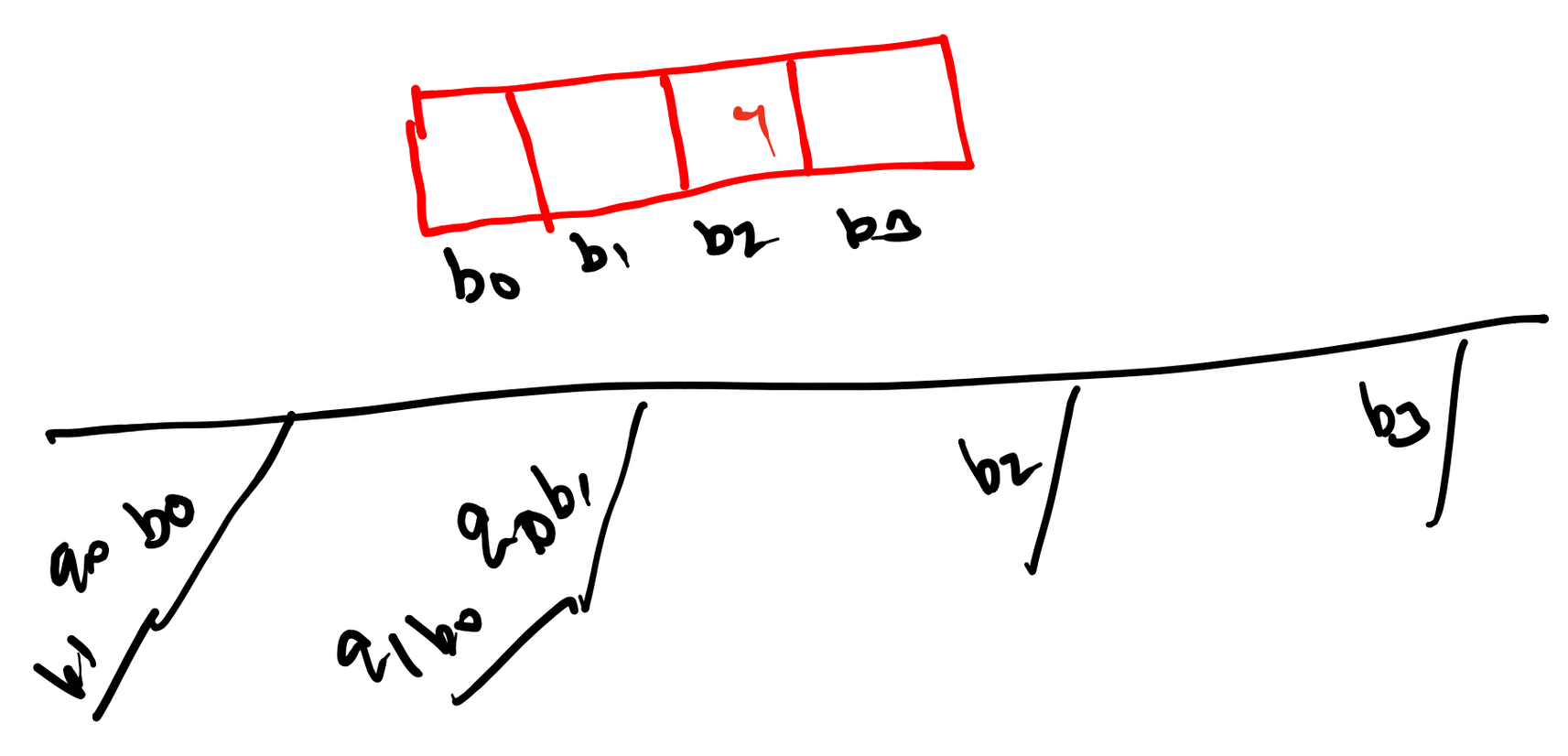


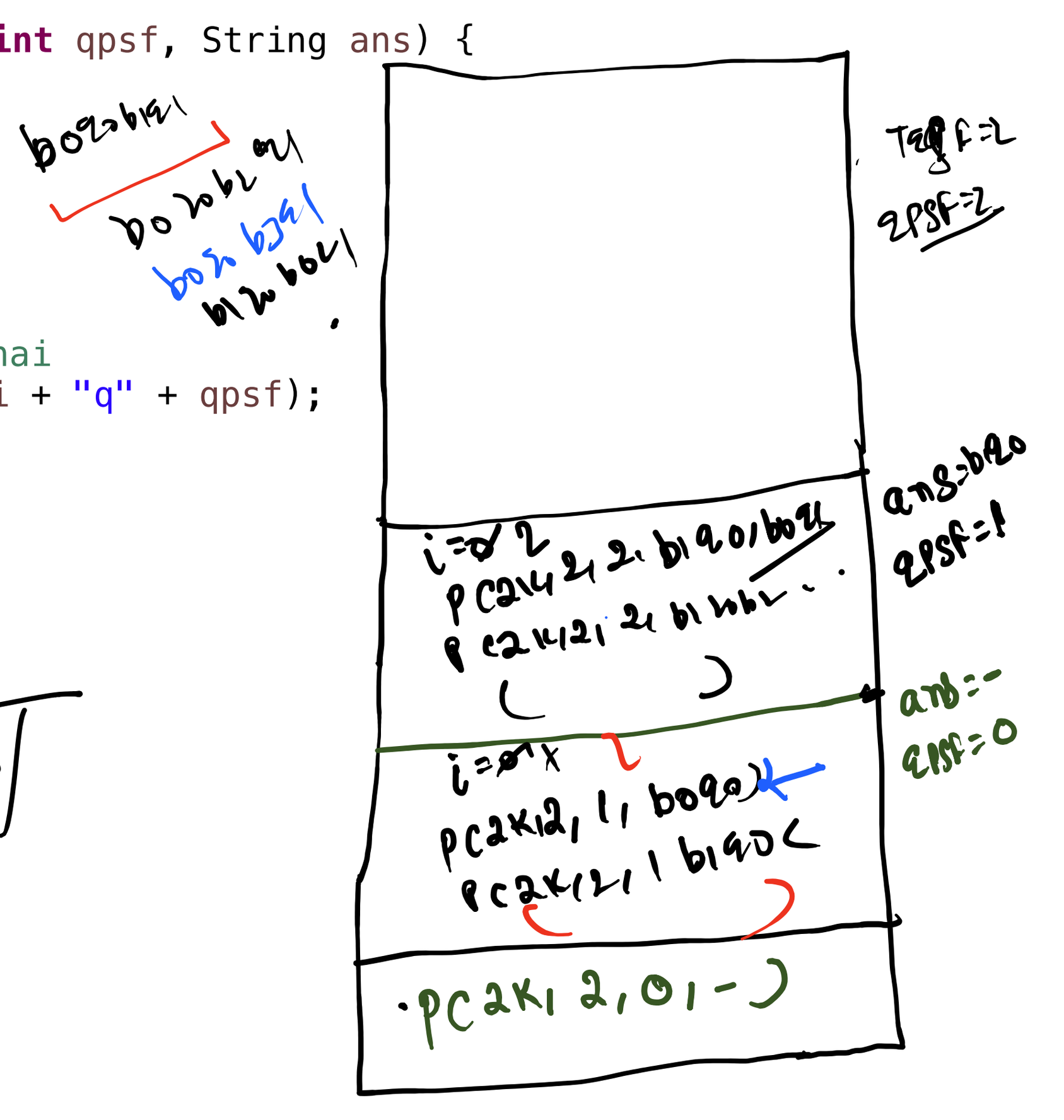


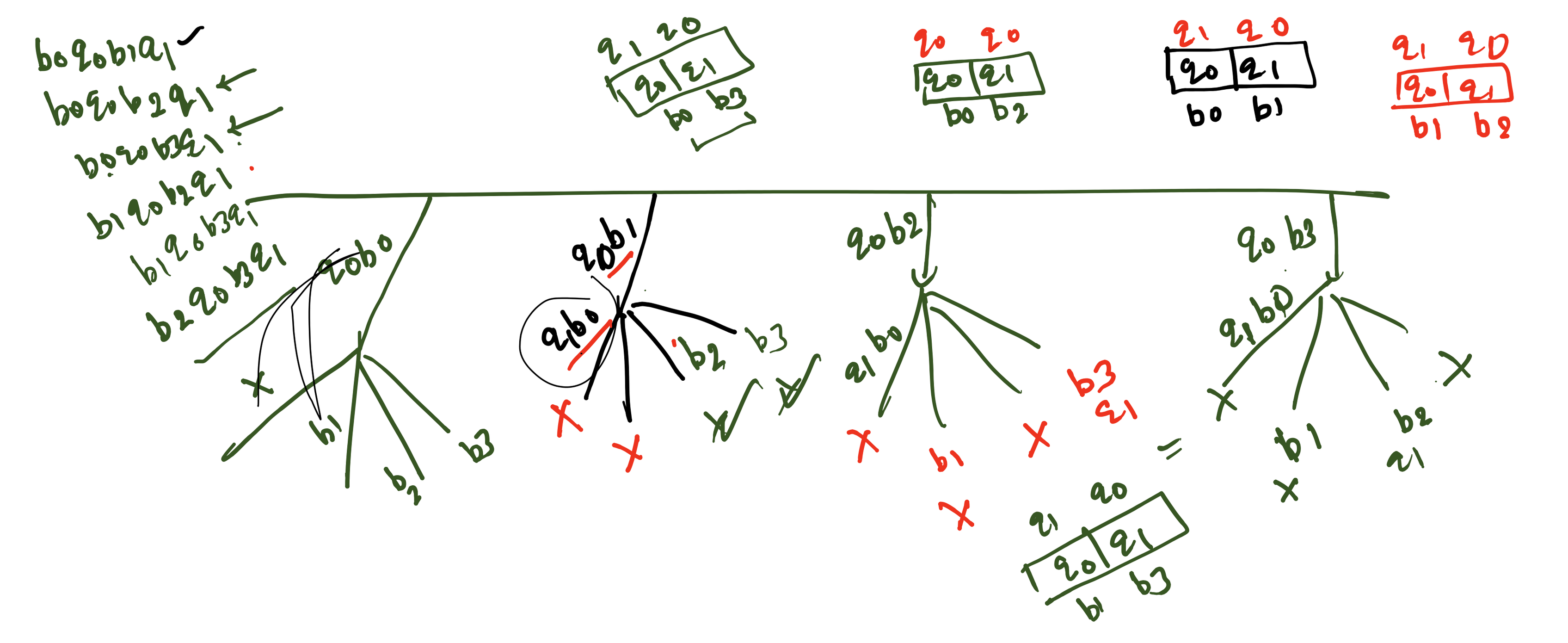
```java
public class Queen_Permutation {

    public static void Permutation(boolean[] board, int tq, int qpsf, String ans) {
        if (tq == qpsf) {
            System.out.println(ans);
            return;
        }

        for (int i = 0; i < board.length; i++) {
            if (board[i] == false) {
                board[i] = true;// marked kr diya Queen yaha hai
                Permutation(board, tq, qpsf + 1, ans + "b" + i + "q" + qpsf);
                board[i] = false;// UNDO
            }
        }
    }
}
```







```java
public static void Combination(boolean[] board, int tq, int qpsf, String ans,int idx) {
    if (tq == qpsf) {
        System.out.println(ans);
        return;
    }

    for (int i = idx; i < board.length; i++) {
        if (board[i] == false) {
            board[i] = true;// marked kr diya Queen yaha hai
            Combination(board, tq, qpsf + 1, ans + "b" + i + "q" + qpsf,i+1);
            board[i] = false;// UNDO
        }
    }
}
```