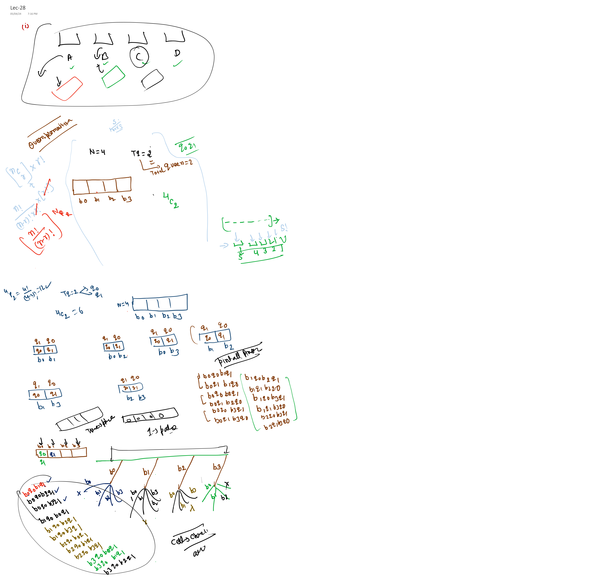
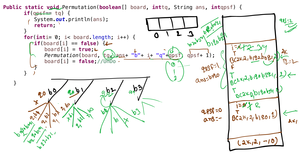
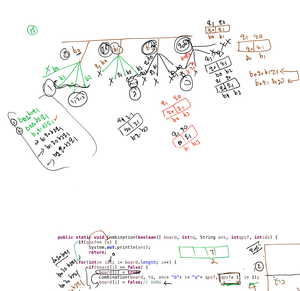
Lec-28
Monday 11:24AM



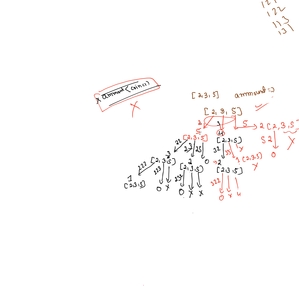
```
public static void main(String[] args) {
    // TODOAuto-generated method stub
    int[] a;
    boolean[] board = new boolean[4];
    int[] p;
    Permutation(board, tq, "", 0);
}

public static void Permutation(boolean[] board, int[] tq, String ans, int[] psf) {
    if(psf[0]==4 ??) {
        System.out.println(ans);
        return;
    }
    for(int i= 0; i< board.length; i++) {
        if(board[i] == false) {
            board[i] = true;
            Permutation(board, tq, ans+ "b"+ i+ "q"+psf[0], psf+ 1);
            board[i] = false;
        }
    }
}
```

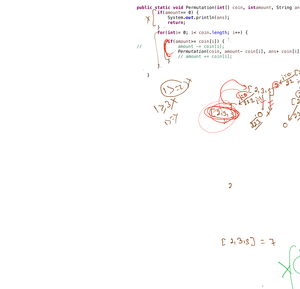


```
Public static void Permutation(boolean[] board, int[] tq, String ans, int[] psf) {
    if(psf[0]==4 ??) {
        System.out.println(ans);
        return;
    }
    for(int i= 0; i< board.length; i++) {
        if(board[i] == false) {
            board[i] = true;
            Permutation(board, tq, ans+ "b"+ i+ "q"+psf[0], psf+ 1);
            board[i] = false;
        }
    }
}
```

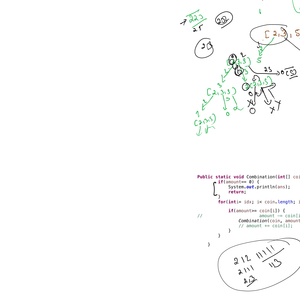


```
public static void Combination(boolean[] board, int[] tq, String ans, int[] psf, int[] is) {
    if(psf[0]==4 ??) {
        System.out.println(ans);
        return;
    }
    for(int i= 0; i< board.length; i++) {
        if(board[i] == false) {
            board[i] = true;
            Combination(board, tq, ans+ "b"+ i+ "q"+psf[0], psf+ 1, is);
            board[i] = false;// TODO
        }
    }
}
```





```
public static void Permutation(int[] coin, int amount, String ans) {
    if(amount== 0) {
        System.out.println(ans);
        return;
    }
    for(int i= 0; i< coin.length; i++) {
        if(amount+= coin[i]) {
            amount += coin[i];
            Permutation(coin, amount- coin[i], ans+ coin[i]);
            // amount -= coin[i];
        }
    }
}
```




```
Public static void Combination(int[] coin, int amount, String ans, int[] is) {
    if(amount== 0) {
        System.out.println(ans);
        return;
    }
    for(int i= idx; i< coin.length; i++) {
        if(amount+= coin[i]) {
            amount += coin[i];
            Combination(coin, amount- coin[i], ans+ coin[i]);
            // amount -= coin[i];
        }
    }
}
```