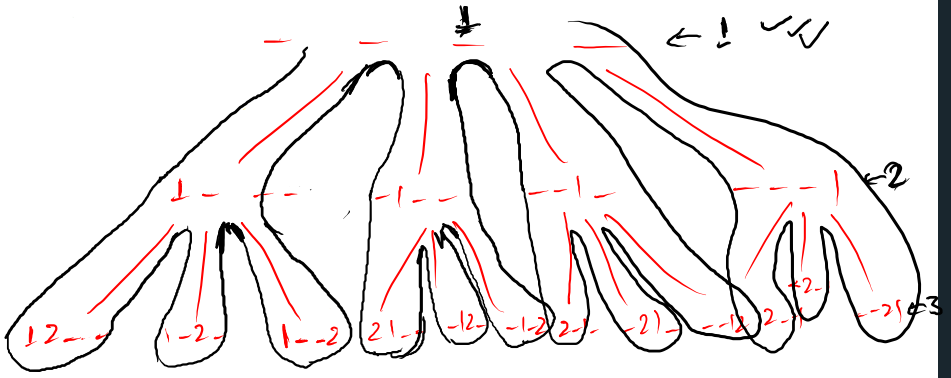
Permutation-1

$^4P_2 = {}^4C_2 \times 12 = \frac{4}{12\ 12} \times 12 = 3 \times 4 = 12$

ti = 2



```
public static void permutations(int[] boxes, int ci, int ti){
  // write your code here

  if(ci > ti){
      for(int i=0;i<boxes.length;i++){
          System.out.print(boxes[i]);
      }

      System.out.println();
      return;
  }


  for(int i=0;i<boxes.length;i++){
      if(boxes[i] == 0){
          boxes[i] = ci;
          permutations(boxes,ci+1,ti);
          boxes[i] = 0;
      }
  }


}
```

1 2 0 0

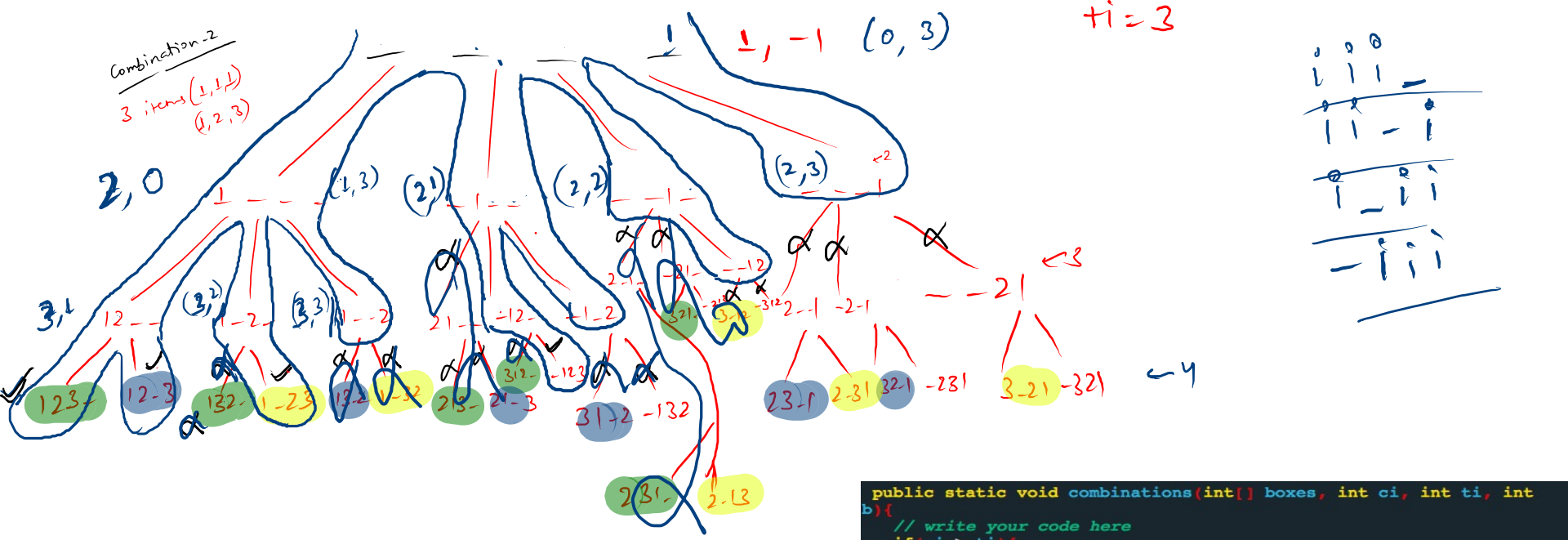
1 0 2 0

1 0 0 2

2 1 0 0

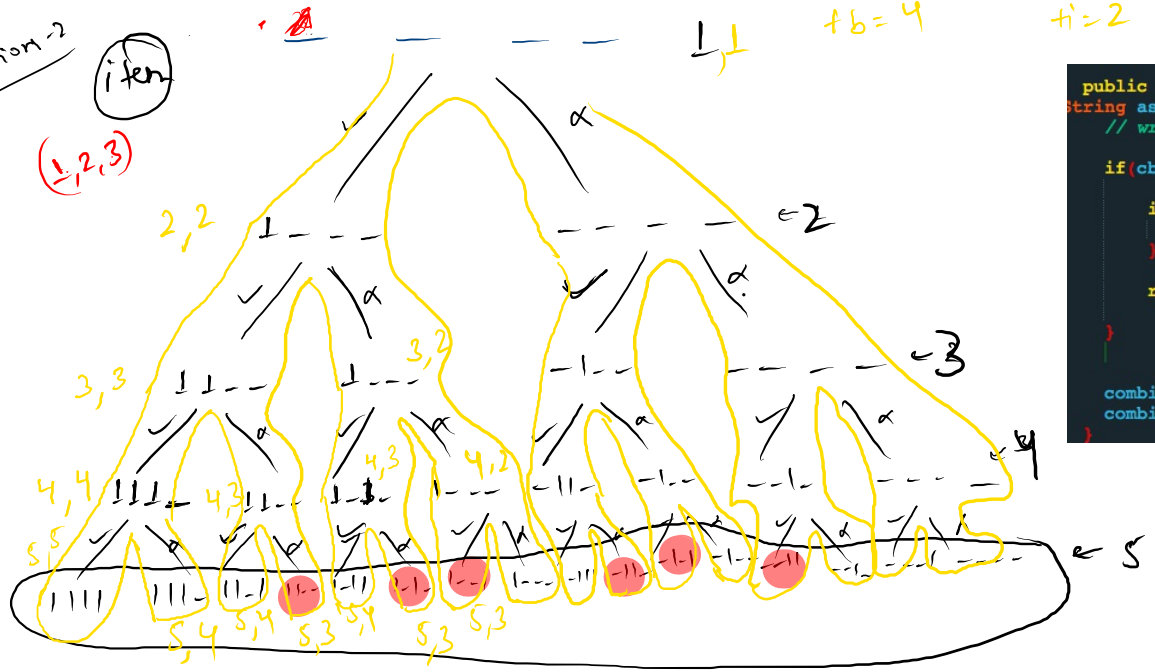
0 1 2 0

0 1 0 2



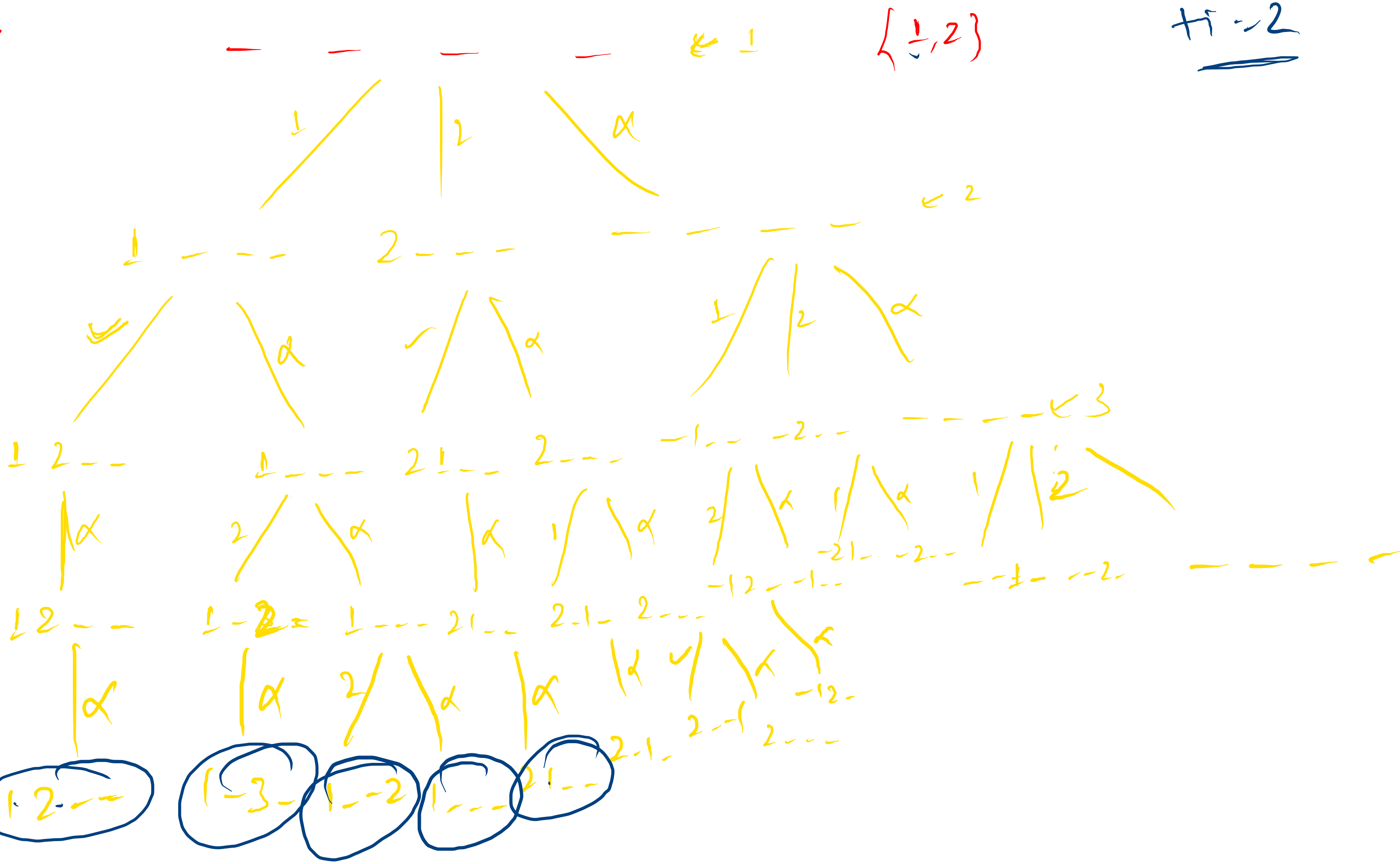

```java
public static void combinations(int[] boxes, int ci, int ti, int lb){
    // write your code here
    if(ci > ti){
        for(int i=0;i<boxes.length;i++){
            if(boxes[i] == 1){
                System.out.print("i");
            }else{
                System.out.print("-");
            }
        }

        System.out.println();
        return;
    }

    for(int i = lb+1;i<boxes.length;i++){
        if(boxes[i] == 0){
            boxes[i] = 1;
            combinations(boxes,ci+1,ti,i);
            boxes[i] = 0;
        }
    }

}
```

Combination-2

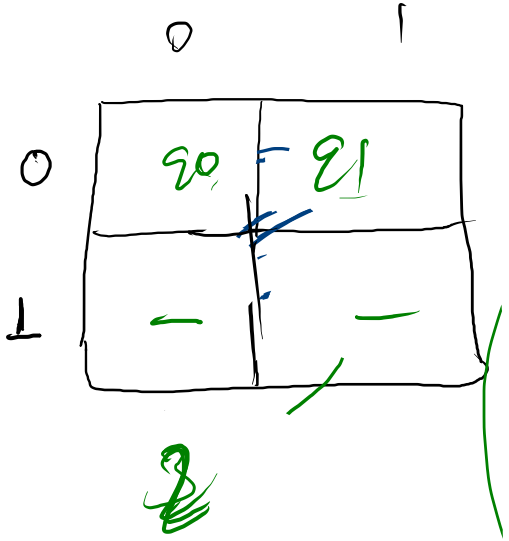


```
public static void combinations(int cb, int tb, int ssf, int ts,
String asf){
   // write your code here

   if(cb > tb){

       if(ssf == ts + 1){
           System.out.println(asf);
       }

       return;

   }


   combinations(cb+1, tb, ssf+1, ts,asf+"i");
   combinations(cb+1, tb, ssf, ts, asf+"-");
}
```

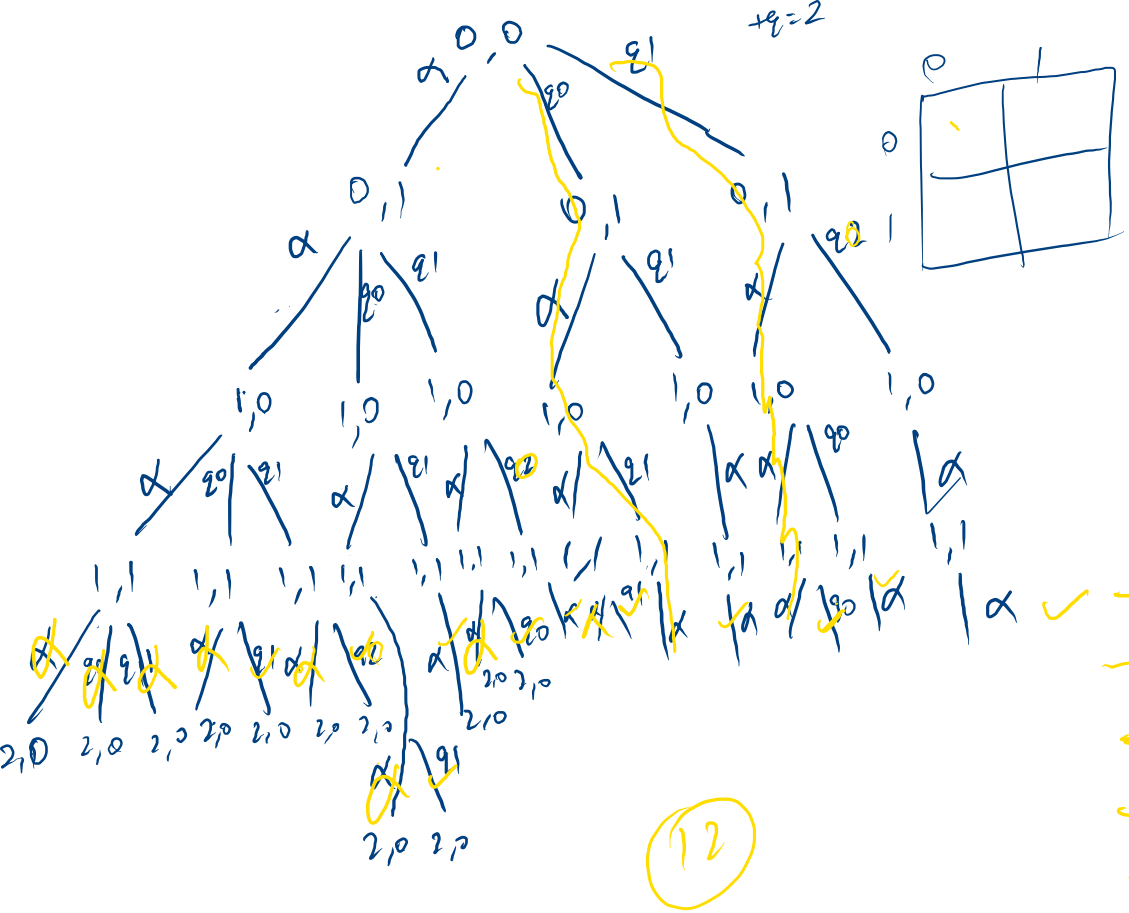
Permutations-2



Queens Comb^n



```java
public static void queensCombinations(int qpsf, int tq, int row, int col, String asf){
    if(row == tq){
        if(qpsf == tq){
            System.out.println(asf);
        }
        return;
    }


    int nxtrow=row;
    int nxtcol = col;
    String yesans = asf;
    String noans = asf;

    if(col == tq-1){
        nxtrow++;
        nxtcol =0;
        yesans = yesans + "q\n";
        noans = noans+"-\n";
    }else{
        nxtcol++;
        yesans = yesans + "q";
        noans = noans +"-";
    }

    queensCombinations(qpsf+1,tq,nxtrow,nxtcol ,yesans);
    queensCombinations(qpsf,tq,nxtrow,nxtcol,noans);
}
```



```
        int nxtrow = row;
        int nxtcol = col;
        char ch = 'a';

        if(col == tq - 1){
            nxtrow++;
            nxtcol = 0;
            ch = '\n';

        }else{
            nxtcol++;
            ch = '\t';
        }

        for(int i = 0;i<queens.length;i++){
            if(queens[i] == false){
                queens[i] = true;
                queensPermutations(qpsf+1,tq,nxtrow,nxtcol,asf + "q" + (i+1
+ ch ,queens);
                queens[i] = false;
            }
        }


        queensPermutations(qpsf,tq,nxtrow,nxtcol,asf+'-'+ch ,queens);
    }
```