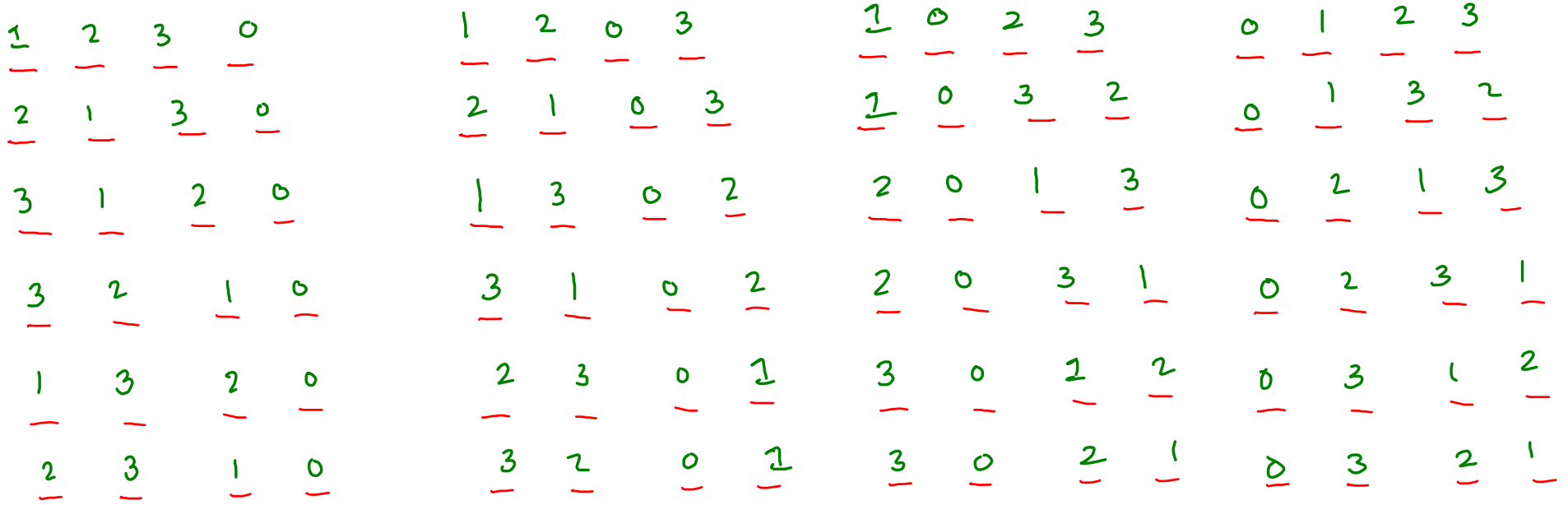
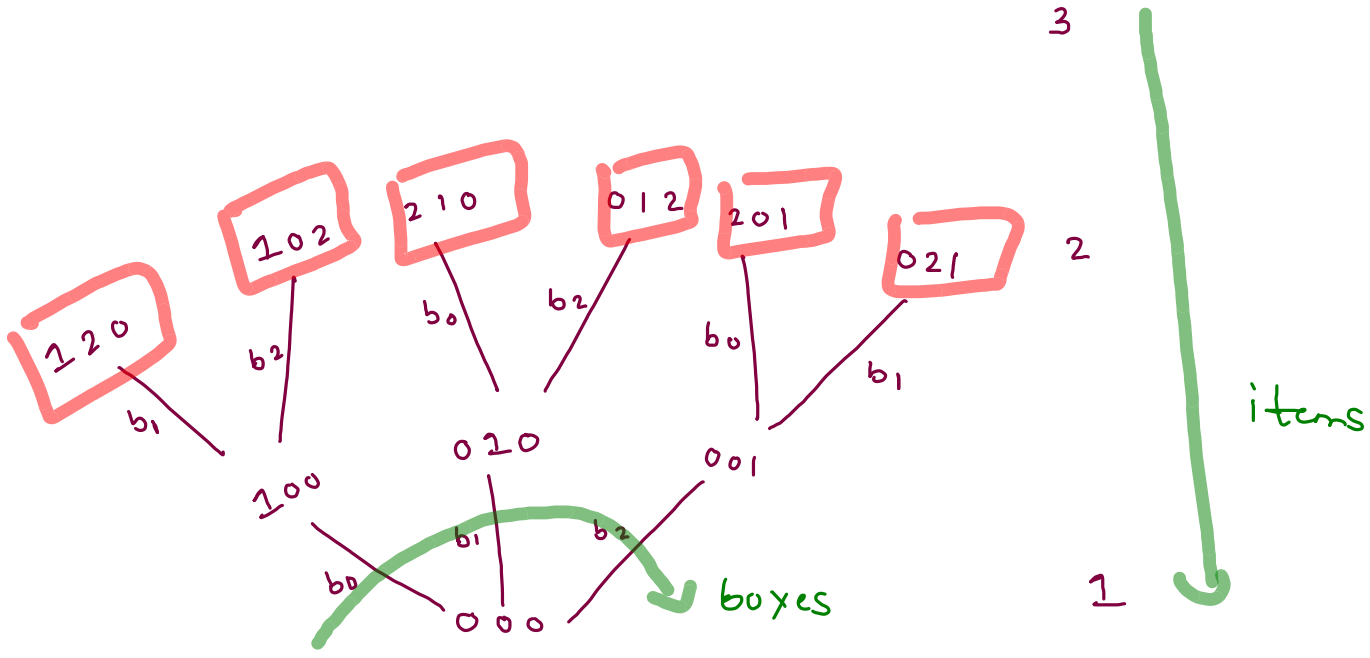
$n = 4$

$r = 3$

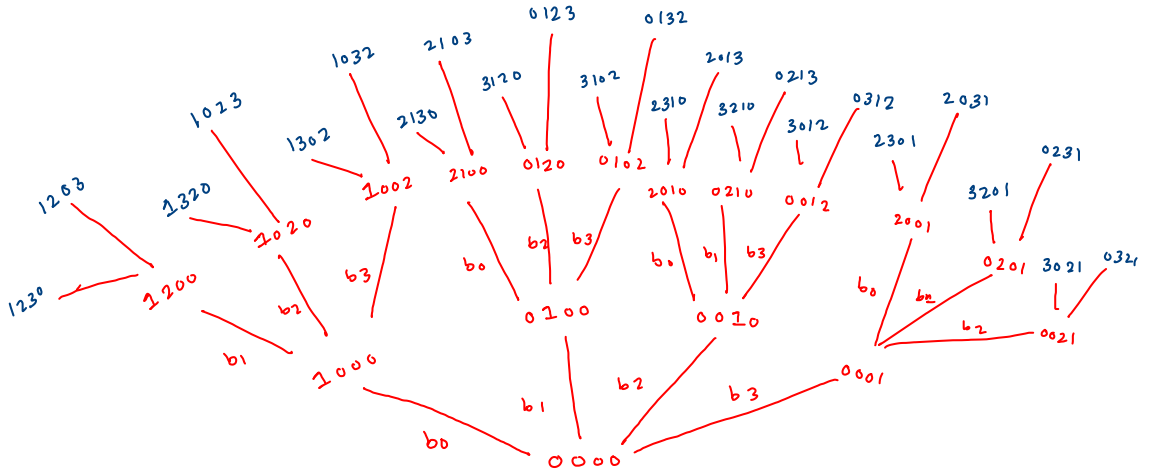

1 2 3 0
2 1 3 0
3 1 2 0
3 2 1 0
1 3 2 0
2 3 1 0
1 2 0 3
2 1 0 3
1 3 0 2
3 1 0 2
2 3 0 1
3 2 0 1
1 0 2 3
1 0 3 2
2 0 1 3
2 0 3 1
3 0 1 2
3 0 2 1
0 1 2 3
0 1 3 2
0 2 1 3
0 2 3 1
0 3 1 2
0 3 2 1

$n = 3$

$r = 2$


120
102
210
012
201
021
100
020
001
000
b1
b2
b0
b2
b0
b2
b0
b1
b1
b2
b0
boxes
3
2
1
itens

$^4P_3 = \frac{4!}{1} = 24 \qquad n = 4,\ r = 3$




$$^nP_r = \frac{n!}{(n-r)!}$$

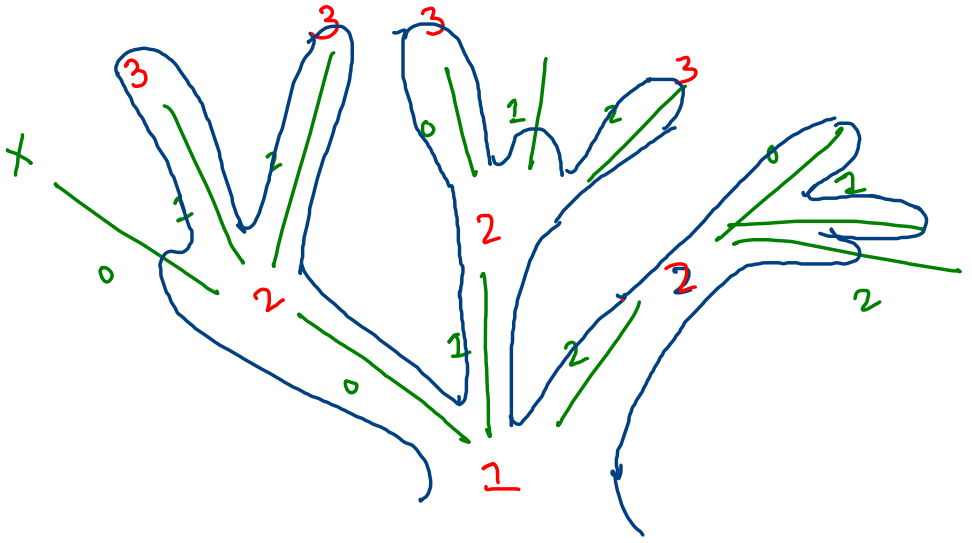
$$^7P_3 = \frac{7 \times 6 \times 5 \times 4!}{4!} = n \times (n-1) \times (n-2) \cdots (n-r+1) \times \frac{(n-r)\cdots 1}{(n-r)!}$$

```java
public static void permutations(int[] boxes, int ci, int ti){
  if(ci > ti) {
      for(int i=0; i < boxes.length;i++) {
          System.out.print(boxes[i]);
      }
      System.out.println();
      return;
  }

  //ci is exploring box choices
  for(int b=0; b < boxes.length;b++) {
      if(boxes[b] == 0) {
          boxes[b] = ci;
          permutations(boxes,ci+1,ti);
          boxes[b] = 0;
      }
  }
}
```

n = 3

ti = 2

120

102

210

012

201

021

$n = 3$ , $r = 2$

| permutations | combinations |
| --- | --- |
| • items → non-identical (distinct) | • items → identical i |
| 1 2 0<br>1 0 2<br>0 1 2<br>2 1 0<br>2 0 1<br>0 2 1 | i i o<br>i o i<br>o i i |
| $^{n}P_{r} = \frac{n!}{(n-r)!}$ | $^{n}C_{r} = \frac{n!}{r!\,(n-r)!}$ |

$$^{n}P_{r} = {}^{n}C_{r} * r!$$

$n = 4$

$r = 3$

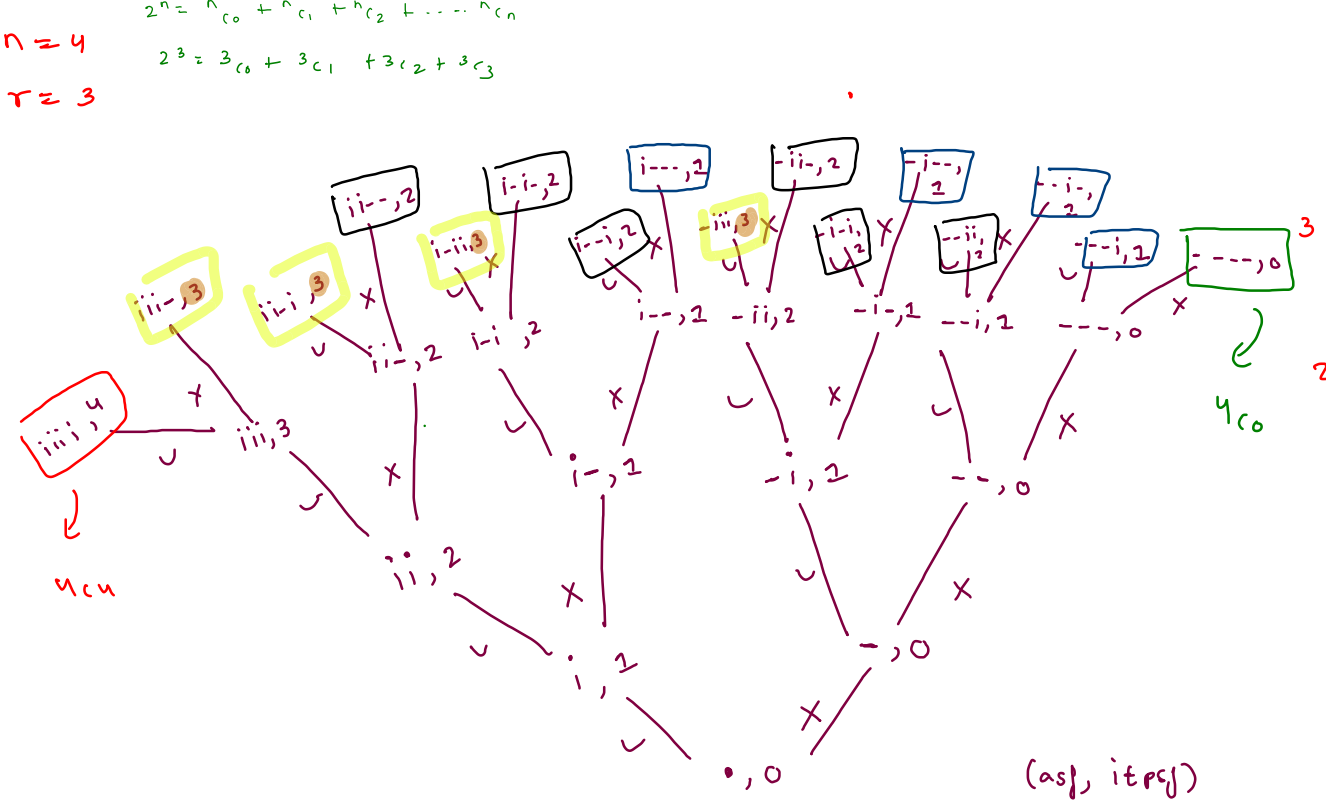
$$^{4}P_{3} = {}^{4}C_{3} * 3!$$

$n = 4$

$r = 3$

$2^n = {}^nC_0 + {}^nC_1 + {}^nC_2 + \cdots {}^nC_n$

$2^3 = {}^3C_0 + {}^3C_1 + {}^3C_2 + {}^3C_3$




$\longrightarrow 4_{C_4} = 1$

$\longrightarrow 4_{C_1} = 4$

$\longrightarrow 4_{C_2} = 6$

$\longrightarrow 4_{C_3} = 4$

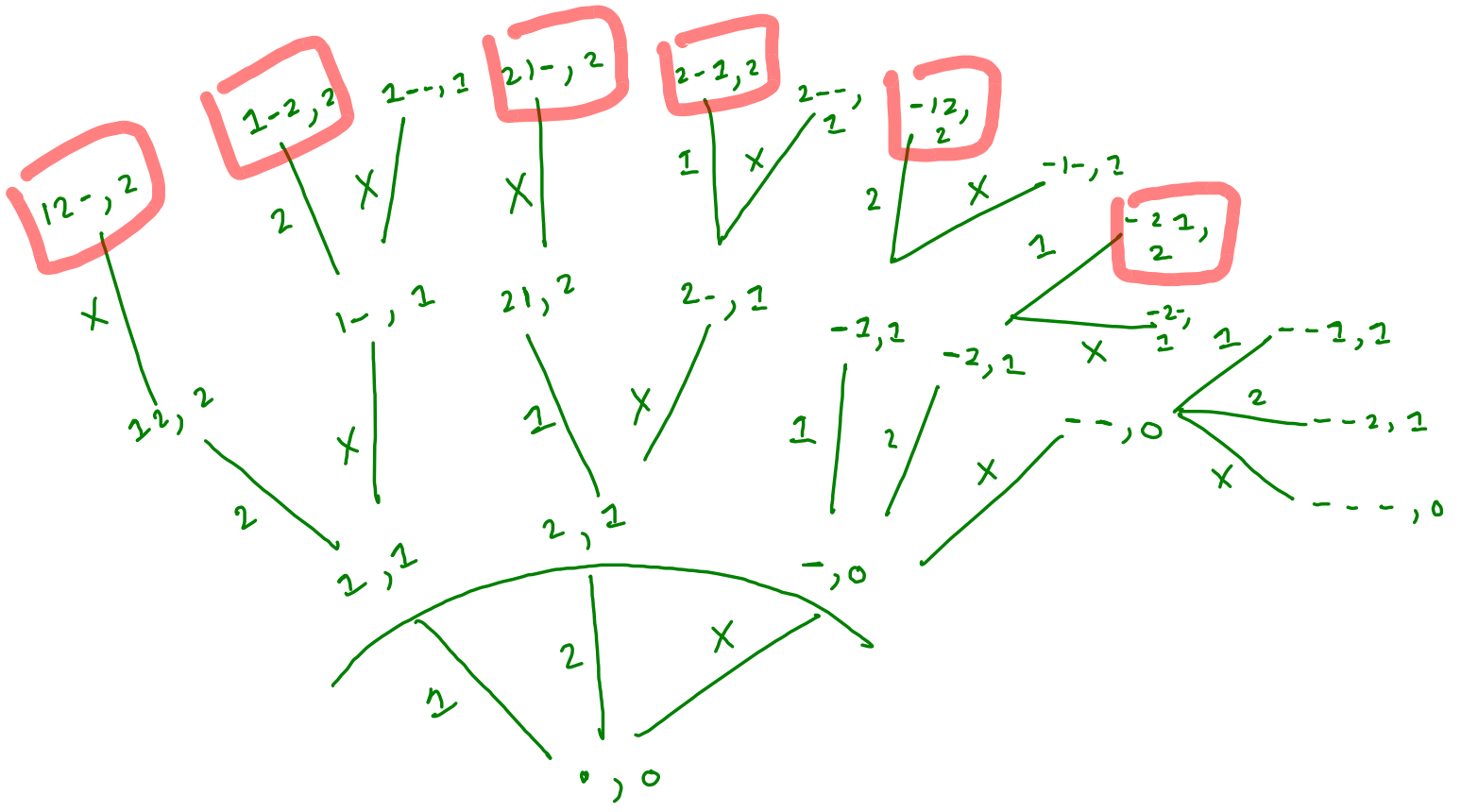
$\longrightarrow 4_{C_0} = 1$

$$2^n = {}^nC_0 + {}^nC_1 + \cdots {}^nC_n$$

$$2^4 = 4_{C_0} + 4_{C_1} + 4_{C_2} + 4_{C_3} + 4_{C_4}$$

P2

$n = 3, \quad r = 2$


12-, 2
1-2, 2
2--, 1
21-, 2
2-1, 2
2--, 2
-12, 2
-1-, 2
-21, 2
-2-, 1
1--, 2
--, 0
---, 0
---, 1
12, 2
1-, 1
21, 2
2-, 1
-1, 2
-2, 2
2, 1
2, 1
-, 0
•, 0
3
2
1

```
public static void permutations(int cb, int tb, boolean[] items, int itpsf, int ts, String asf)
```

120

102

210

201

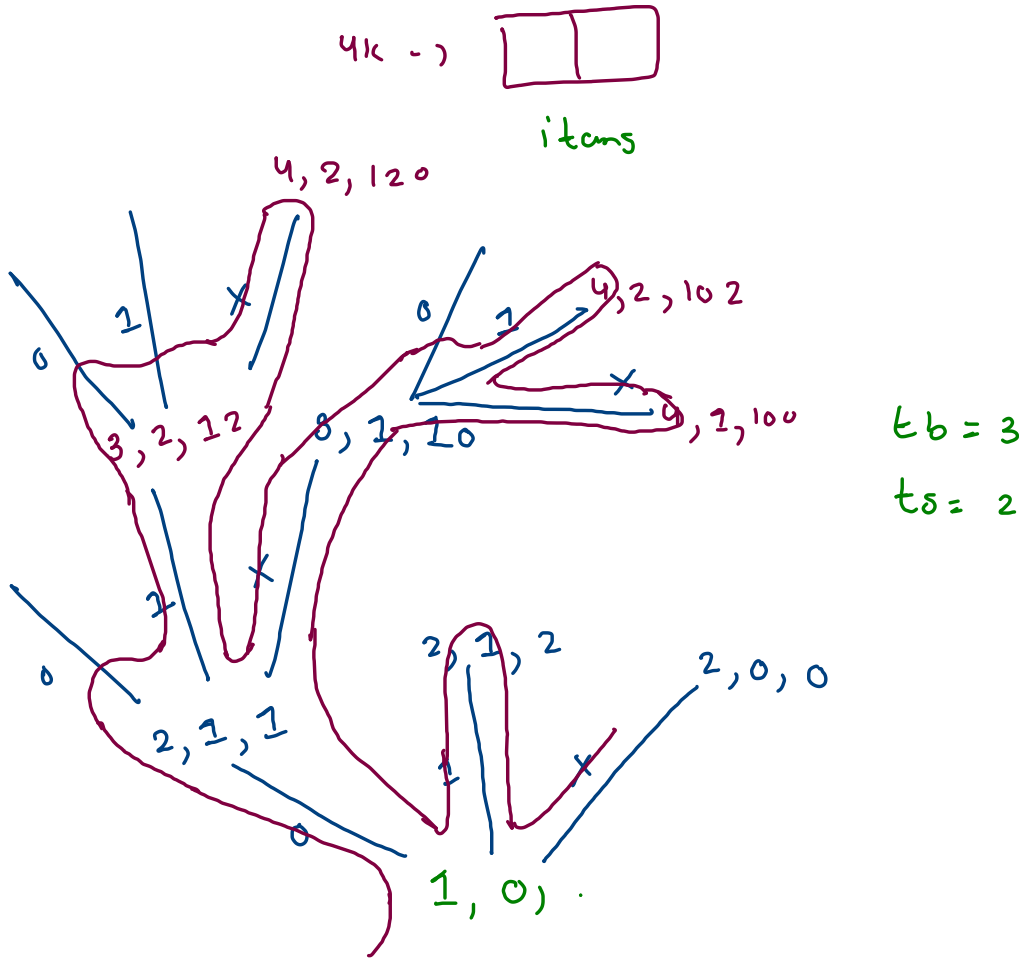
012

021

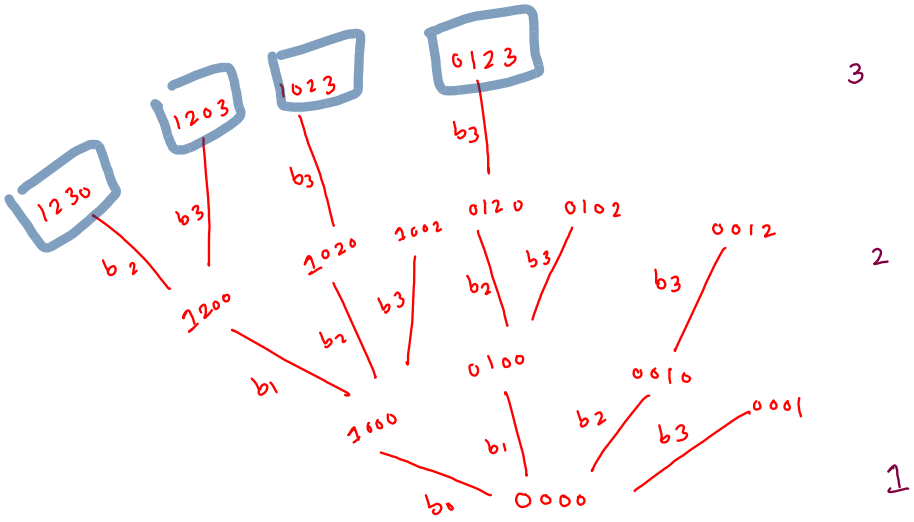
```
if(cb > tb) {
    if(itpsf == ts) {
        System.out.println(asf);
    }
    return;
}

//yes calls -> which item to select
for(int it = 0; it < items.length;it++) {
    if(items[it] == false) {
        items[it] = true;
        permutations(cb+1,tb,items,itpsf+1,ts,asf+(it+1));
        items[it] = false;
    }
}

//no call
permutations(cb+1,tb,items,itpsf,ts,asf+0);
```



n = 4 , r = 3




$b_0, b_2, b_3$

n = 4

r = 3

1 X 2 3

1 0 2 3

1 0 3 2

2 0 1 3

2 0 3 1

3 0 2 1

3 0 1 2

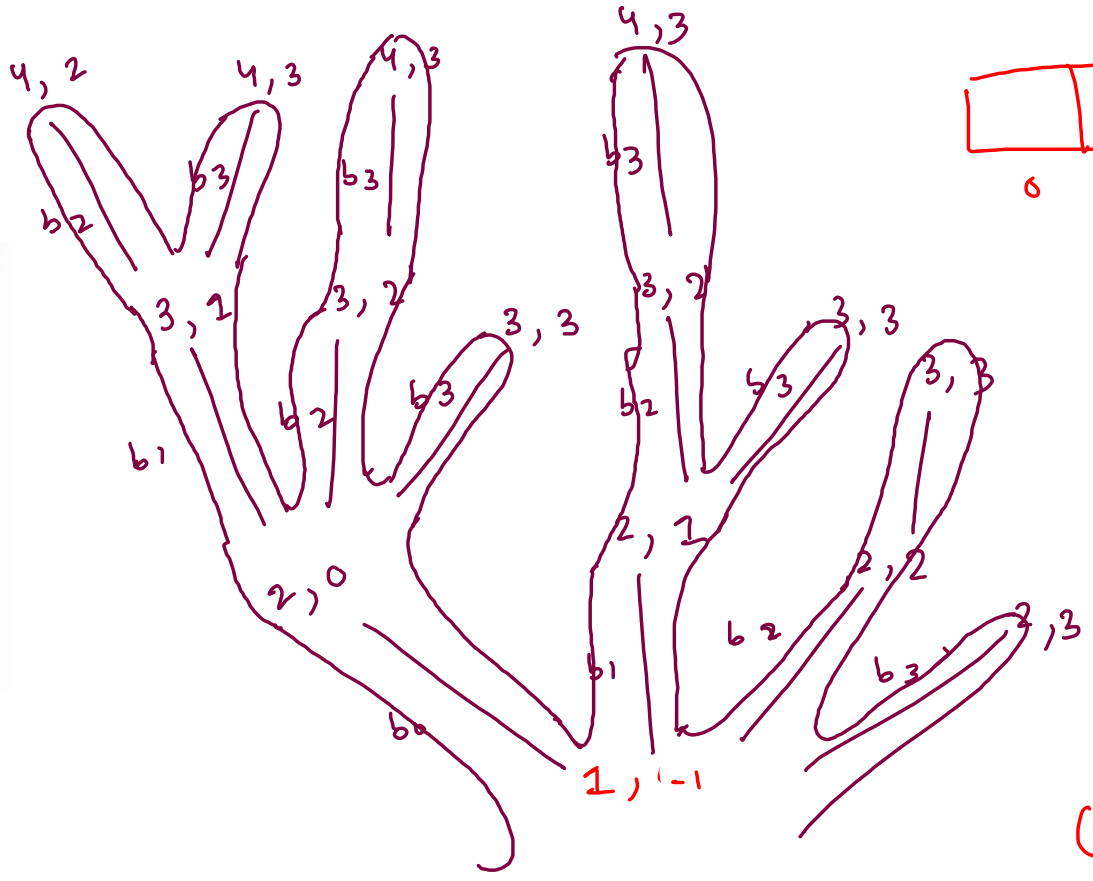
n = 4

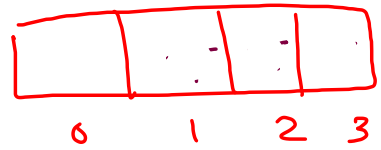
r = 3

```
public static void combinations(boolean[] boxes, int ci, int ti, int lb){

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

  for(int b = lb+1; b < boxes.length;b++) {
      if(boxes[b] == false) {
          boxes[b] = true;
          combinations(boxes,ci+1,ti,b);
          boxes[b] = false;
      }
  }
}
```




iiio

iioi

ioii

oiii

(ci, lb)