Permutation - 1

↳ arrangement

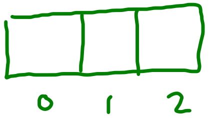
$$^{n}P_{r} = \frac{n!}{(n-r)!}$$

$n = 3$

$i = 2$

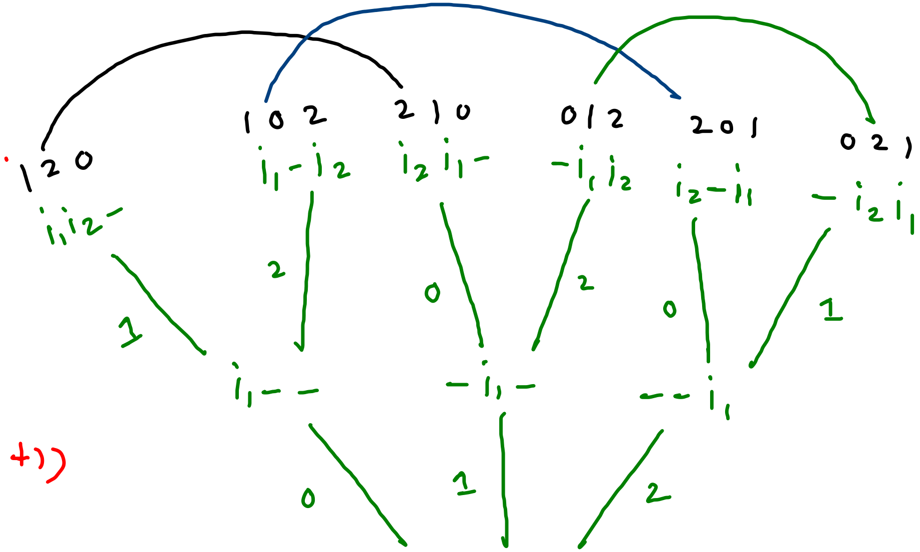
$i_1 \quad i_2$

120	i_1	i_2	0	i_2	i_1	0	210
102	i_1	0	i_2	i_2	0	i_1	201
012	0	i_1	i_2	0	i_2	i_1	021



$n = 3$

$r = 2$



i_2

i_1

$^nP_r = n * (n-1)(n-2) \ldots \ldots (n-r+1)$

```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;
    }

    //boxes - options
    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 boxes

r = 2 items

1 2 0

1 0 2

2 1 0

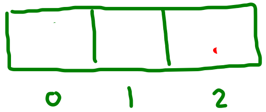
0 1 2

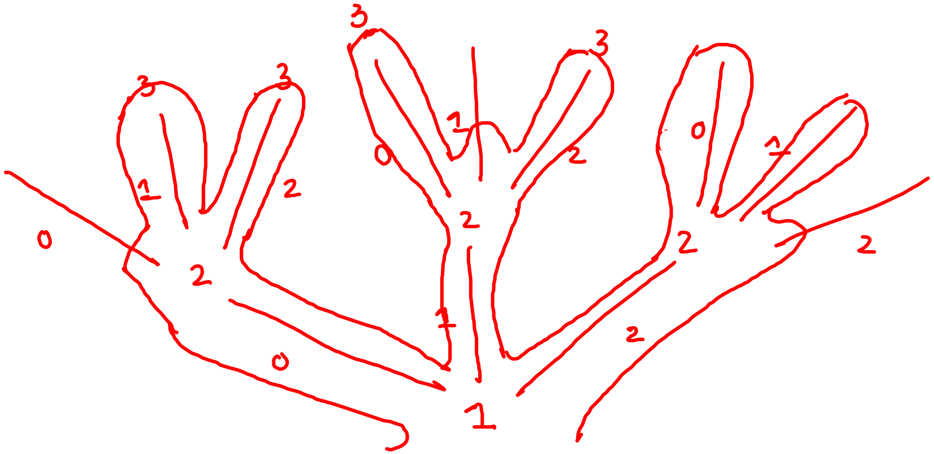
2 0 1

0 2 1

ti = 2

boxes





$$^nP_r = \frac{n!}{(n-r)!} = \frac{n * (n-1) * (n-2) \cdots (n-r+1)(n-r) \cdots 1}{(n-r)!}$$

$$k! = k(k-1)!$$

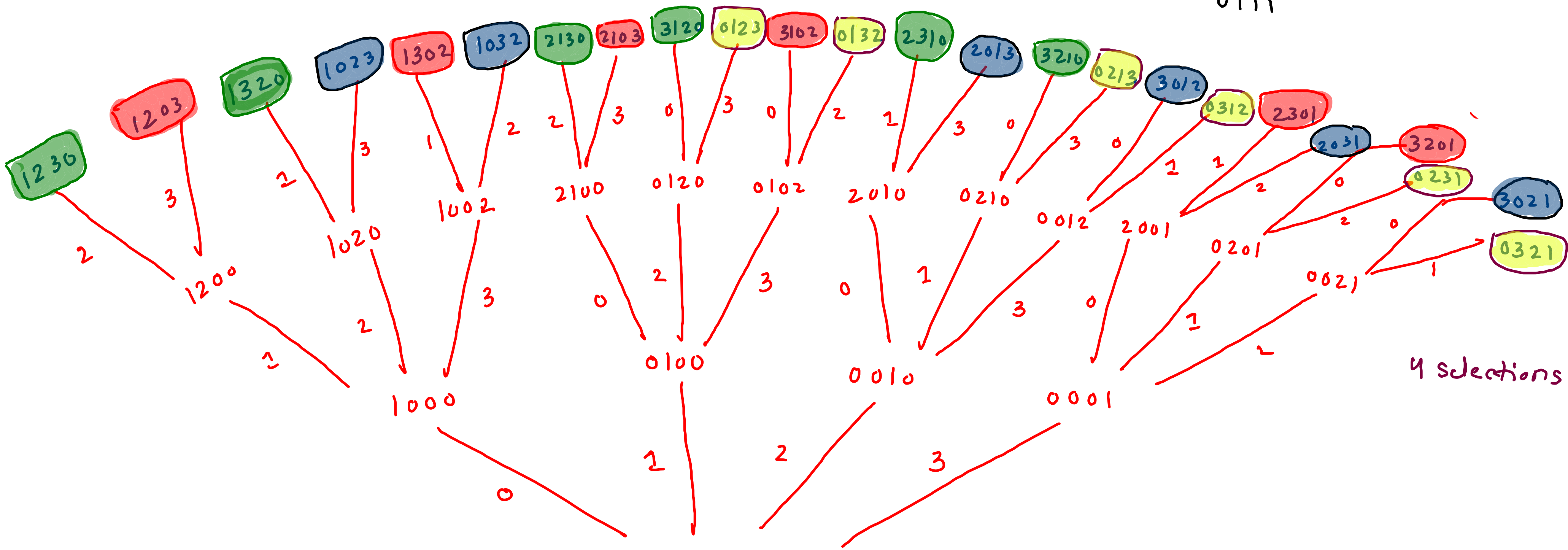
$$= \frac{n * (n-1) * (n-2) \cdots (n-r+1)!}{(n-r)!}$$

$$= \frac{n * (n-1) * (n-2) \cdots (n-r+1) * \cancel{(n-r)!}}{\cancel{(n-r)!}}$$

$$= n * (n-1) * (n-2) \cdots (n-r+1)$$

$^4P_3 = 24, \quad ^4C_3 = 4$

$n = 4, \quad r = 3$



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

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

$$\boxed{^{n}P_{r} = {}^{n}C_{r} \times r!}$$

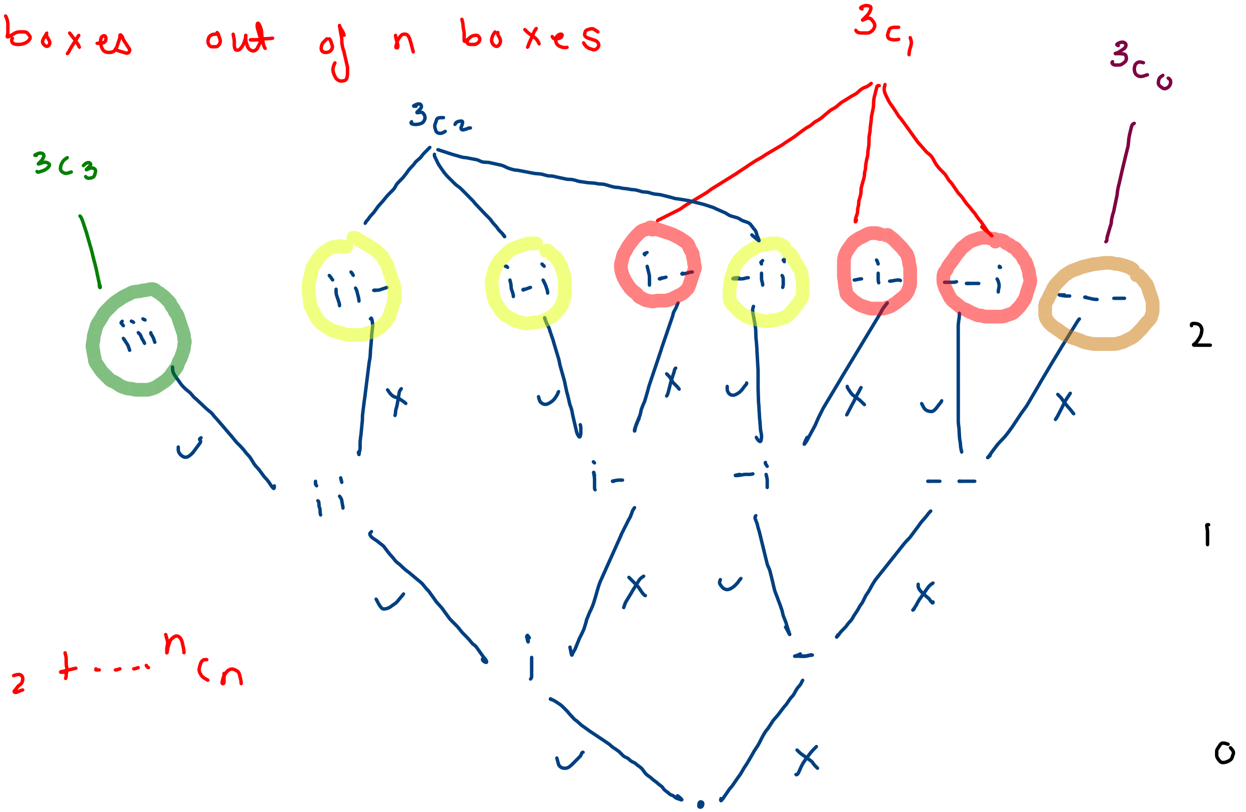
Permutation	combinations
(i) arrangement	(i) selection
(ii) non-identical	(ii) identical items
$n = 3 \quad r = 2$	$n = 3 \quad r = 2$
$i_1 i_2 0 \qquad i_2 i_1 0$ $\longrightarrow$	$i \; i \; 0$
$i_1 0 i_2 \qquad i_2 0 i_1$ $\longrightarrow$	$i \; 0 \; i$
$0 i_1 i_2 \qquad 0 i_2 i_1$ $\longrightarrow$	$0 \; i \; i$

Combinations - 1

$n = 3$

$r = 2$

↳ select r boxes out of n boxes



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

Permutations - 2

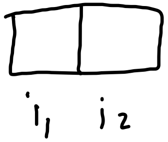
$n = 3$

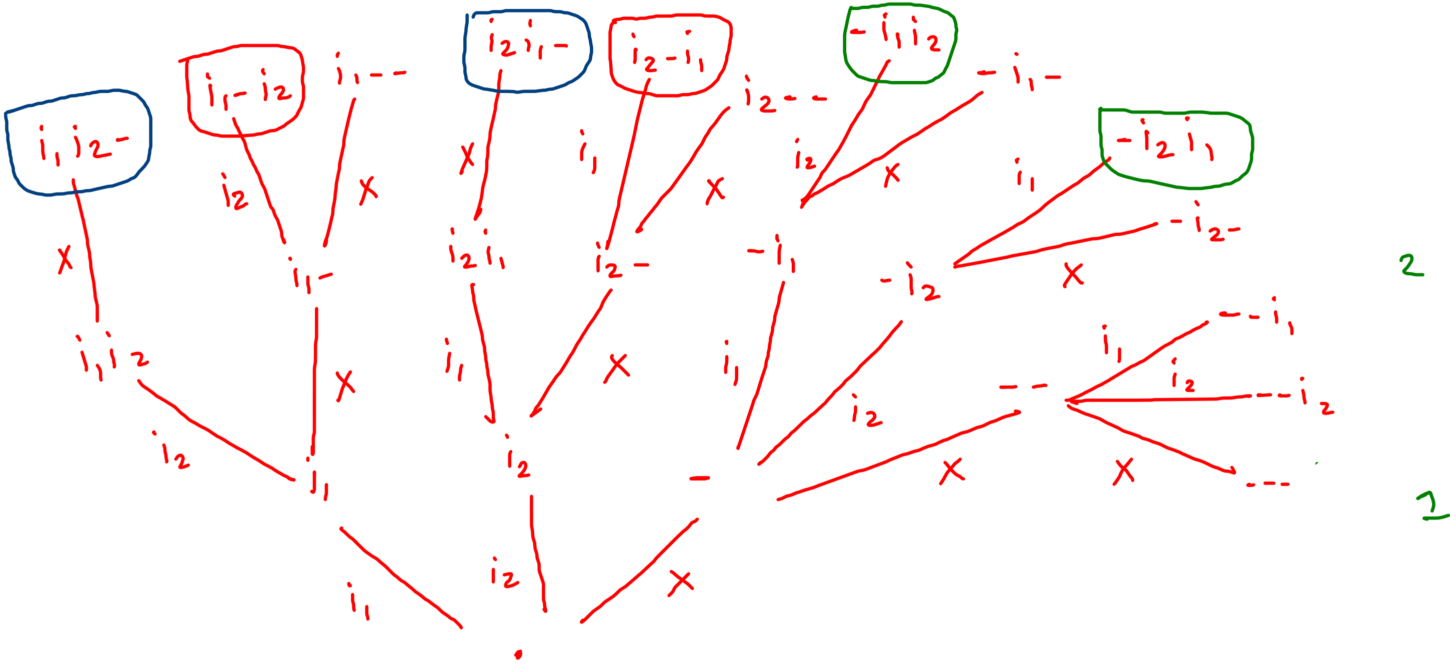
$r = 2$

boxes : level

items : options

(idea : comb 1)

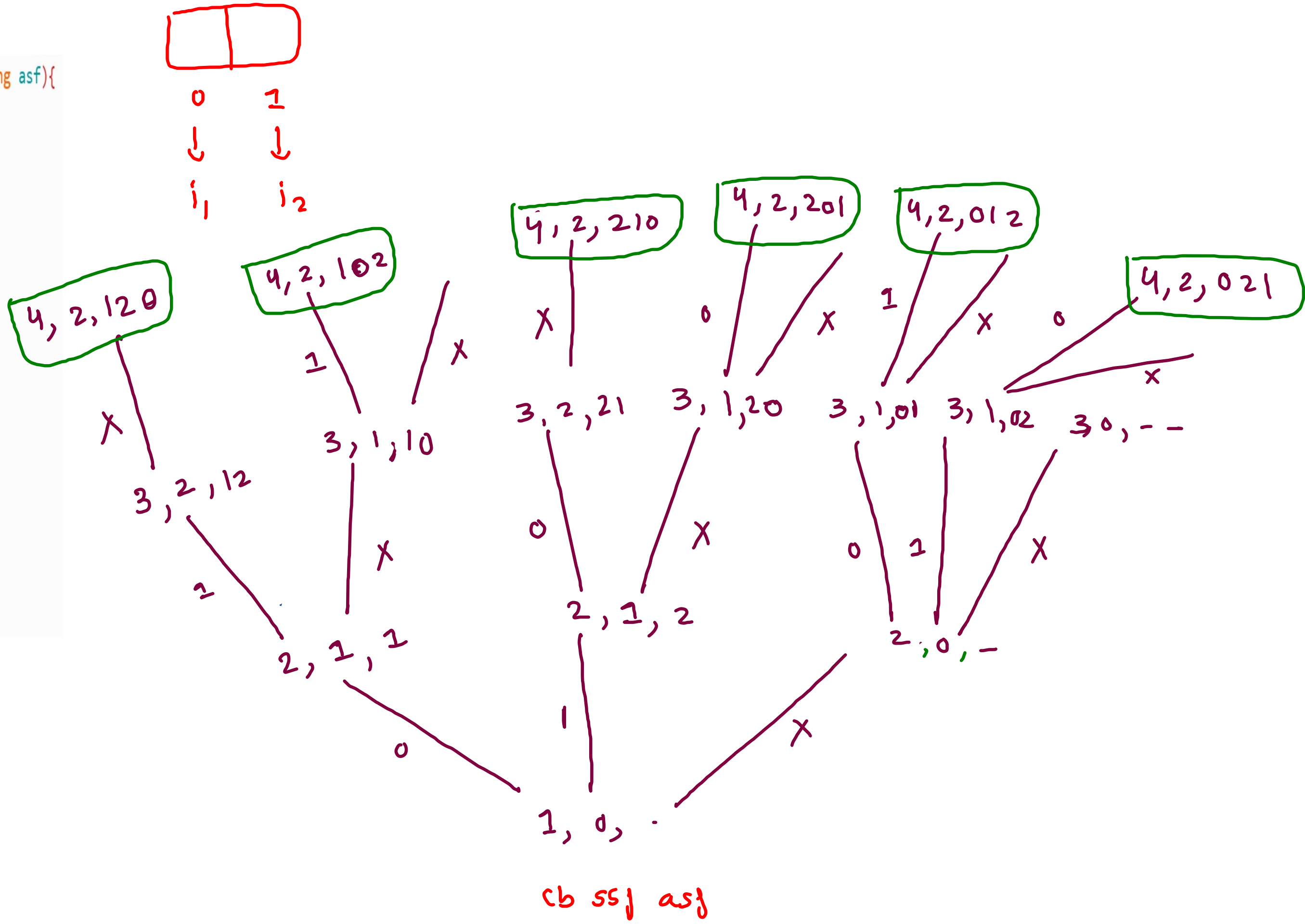




```java
public static void permutations(int cb, int n, int[] items, int ssf, int r, String asf){
  if(cb > n) {
    if(ssf == r) {
      System.out.println(asf);
    }
    return;
  }

  //yes calls - choose an item
  for(int i=0; i < items.length;i++) {
    if(items[i] == 0) {
      items[i] = 1;
      permutations(cb + 1, n,items,ssf + 1,r,asf + (i+1));
      items[i] = 0;
    }
  }

  //no call
  permutations(cb + 1,n,items,ssf,r,asf + "0");
}
```



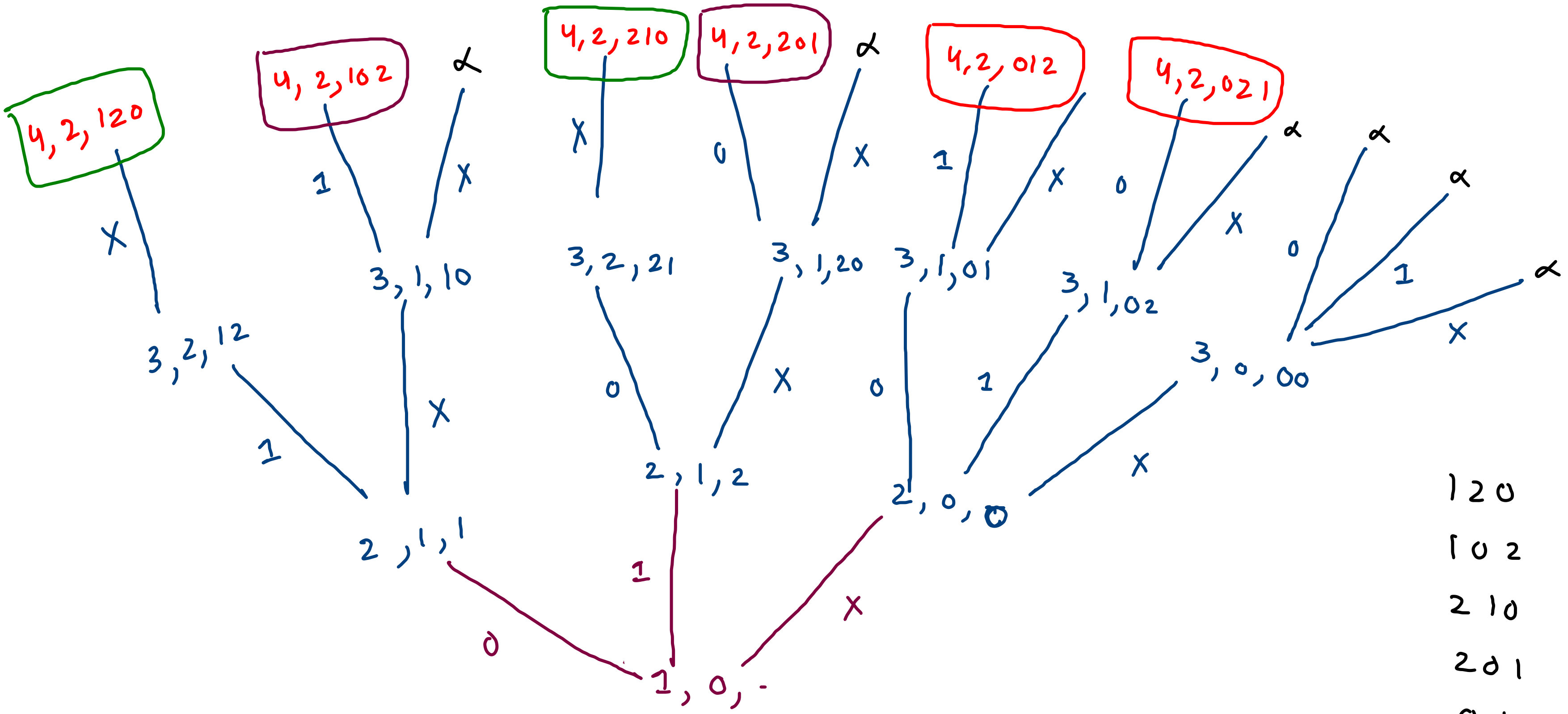
cb, ssf, asf

n = 3

r = 2

$n = 3, \quad r = 2$

cb, ssj, .

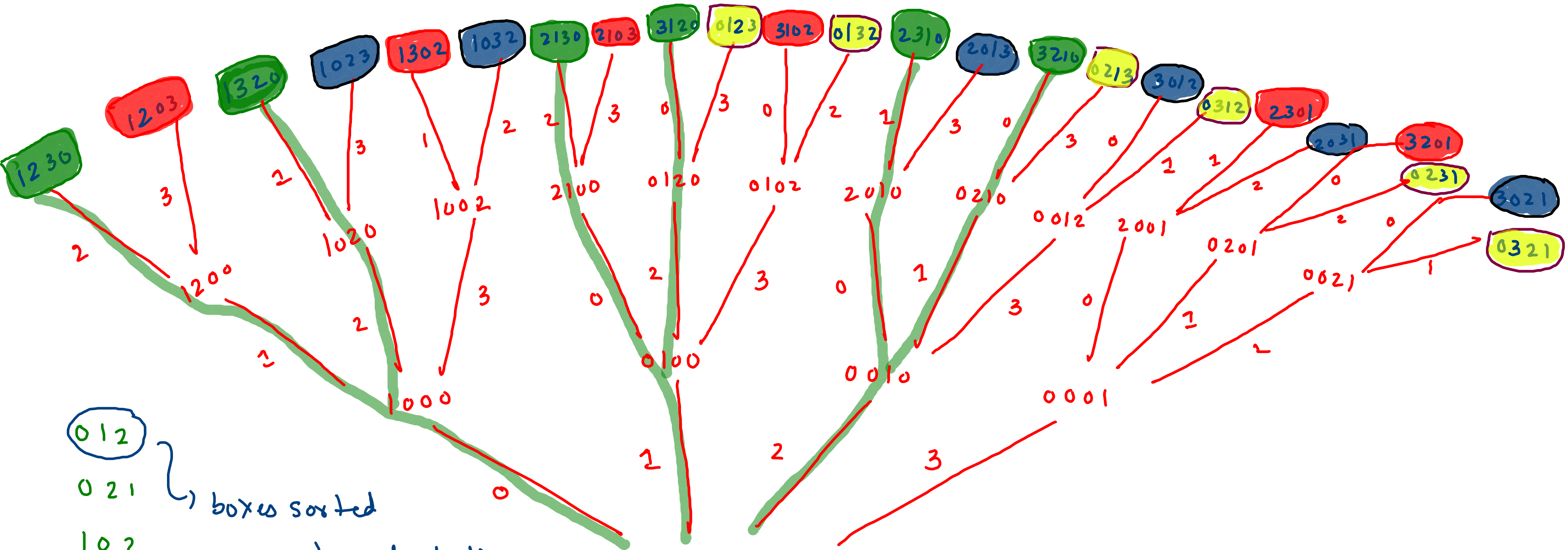


120
102
210
201
012
021

Combinations - 2

$n = 4$

$r = 3$



012
021
102
120
201
210

boxes sorted

order of selection

$r = 3$

			X
0	1	2	3

1 2 3 0

1 3 2 0

2 1 3 0

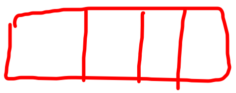
3 1 2 0

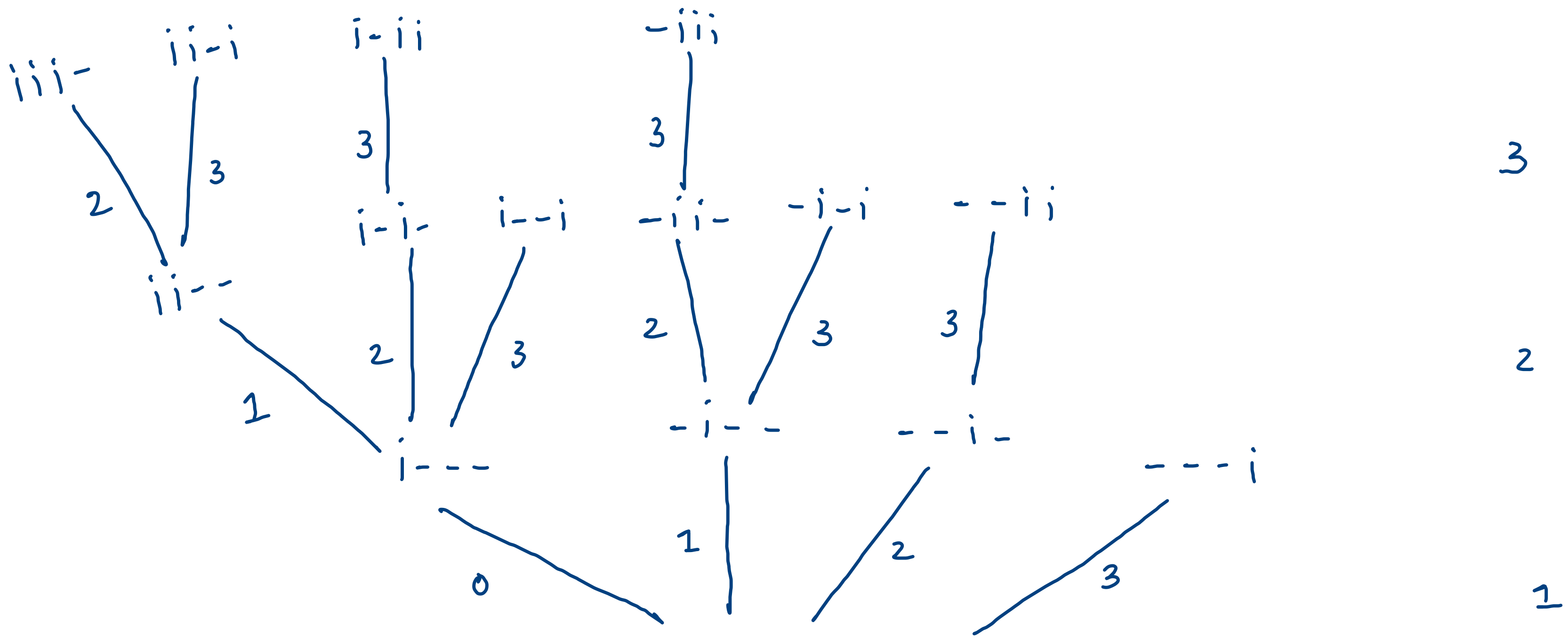
2 3 1 0

3 2 1 0

$\longrightarrow$ 1110

$n = 4 \quad r = 3$





n = 3 , r = 2

```java
public static void combinations(int[] boxes, int ci, int ti, int lb){
    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 b = lb+1; b < boxes.length;b++) {
        boxes[b] = 1;
        combinations(boxes,ci + 1,ti,b);
        boxes[b] = 0;
    }
}
```

4 comparison