

## Permutation - 1

$$n = 4$$

$k = 3$  (all items are distinct)

$$\text{ways} = {}^4P_3 = \frac{4!}{1!} = 24$$

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

$$n = 3$$

$$k = 2$$

$$i_1 i_2 - \quad 1 \ 2 \ 0$$

$$i_1 - i_2 \quad 1 \ 0 \ 2$$

$$- i_1 i_2 \quad 0 \ 1 \ 2$$

$$i_2 i_1 - \quad 2 \ 1 \ 0$$

$$i_2 - i_1 \quad 2 \ 0 \ 1$$

$$- i_2 i_1 \quad 0 \ 2 \ 1$$

$$N = 4$$

$$K = 3$$

$$4P_3 = 24$$

$$4C_3 = 4$$

2	2		3
0	1	2	3

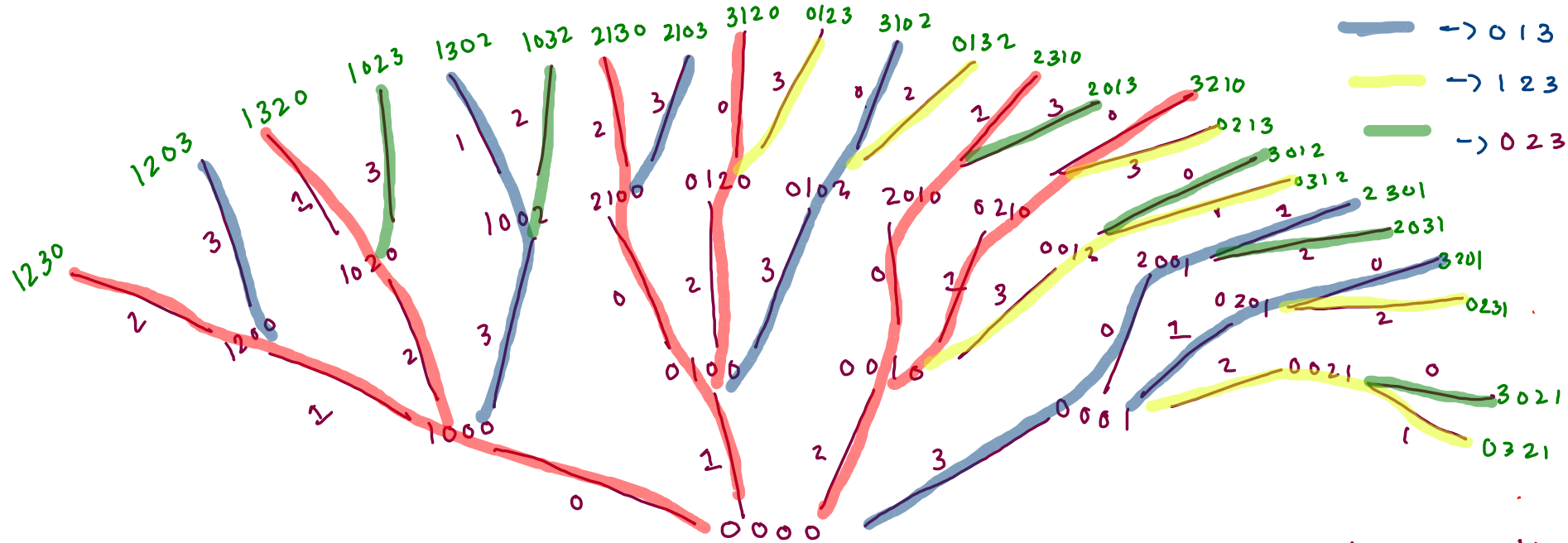
box selected

→ 012

→ 013

→ 123

→ 023



levels → items

options → boxes

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

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

$$\begin{aligned} {}^4P_3 &= {}^4C_3 \times 3! \\ &= 4 \times 6 = 24 \end{aligned}$$

$${}^nP_r = ({}^nC_r) \times r!$$

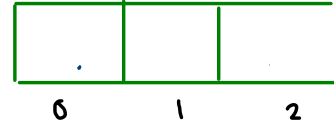
```

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;
    }

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

```

41<



$$n = 3$$

$$r = 2$$

$$ti = 2$$

1 2 0

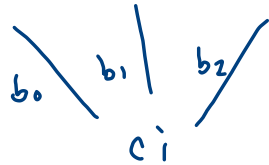
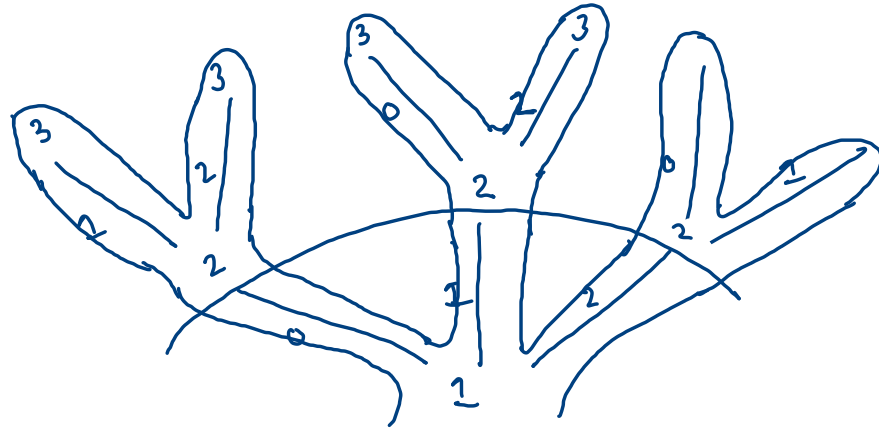
1 0 2

2 1 0

0 1 2

2 0 1

0 2 1



## Permutation

(i) arrangement (distinct items)

$$n = 3, \quad r = 2$$

1 2 0

2 1 0

1 0 2

2 0 1

0 1 2

0 2 1

}

}

}

→

→

→

## Combination

(i) selection (identical items)

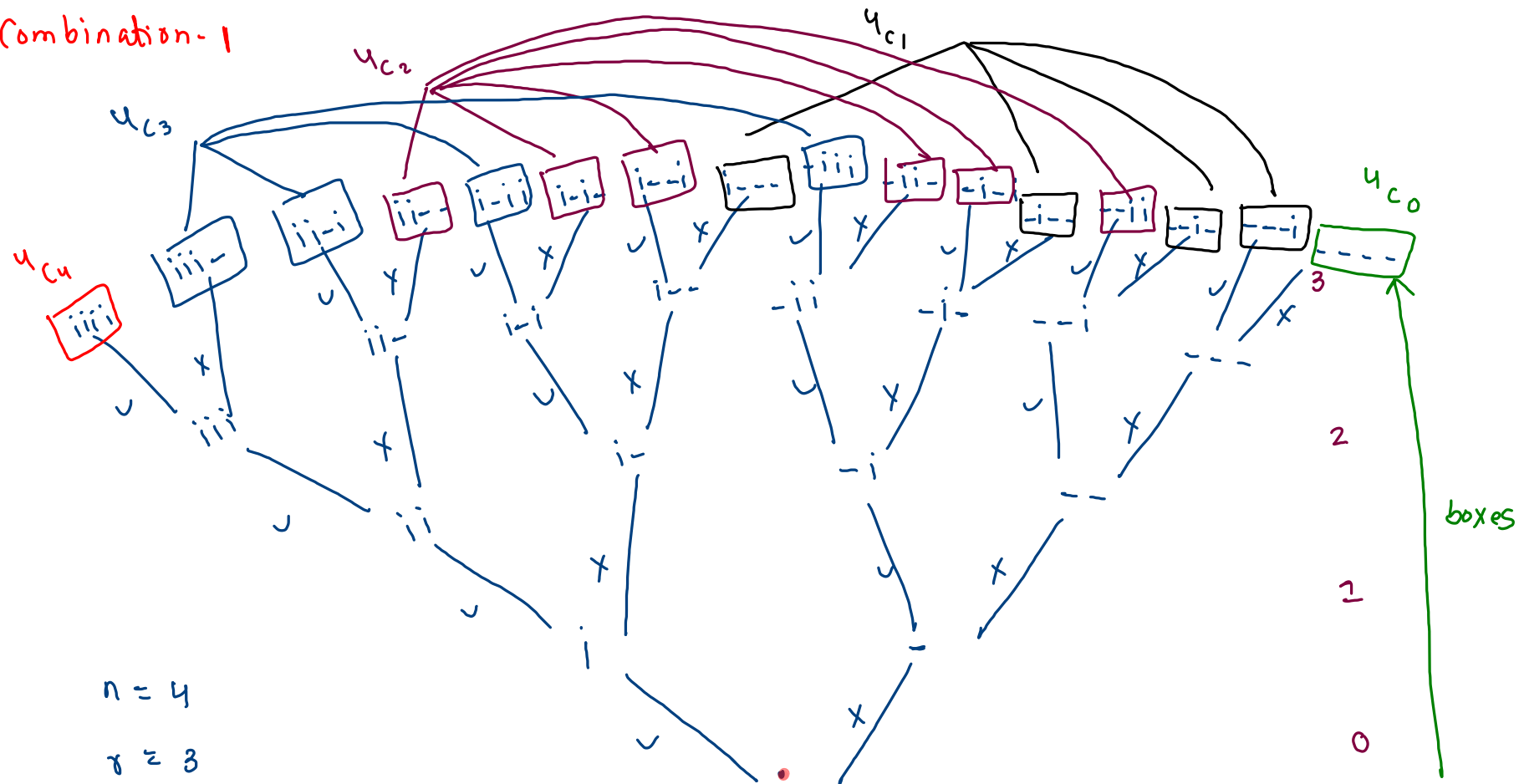
$$n = 3, \quad r = 2$$

i i -

i - i

- i i

Combination-1



$$2^4 = u_{c0} + u_{c1} + u_{c2} + u_{c3} + u_{c4}$$

$$= 1 + 4 + 6 + 4 + 1$$

```

public static void combinations(int cb, int tb, int ssf, int ts, String asf){
    if(cb > tb) {
        if(ssf == ts) {
            System.out.println(asf);
        }
        return;
    }

    //box choice -> yes
    combinations(cb+1, tb, ssf+1, ts, asf + "i");

    //box choice -> no
    combinations(cb+1, tb, ssf, ts, asf + "-");
}

```

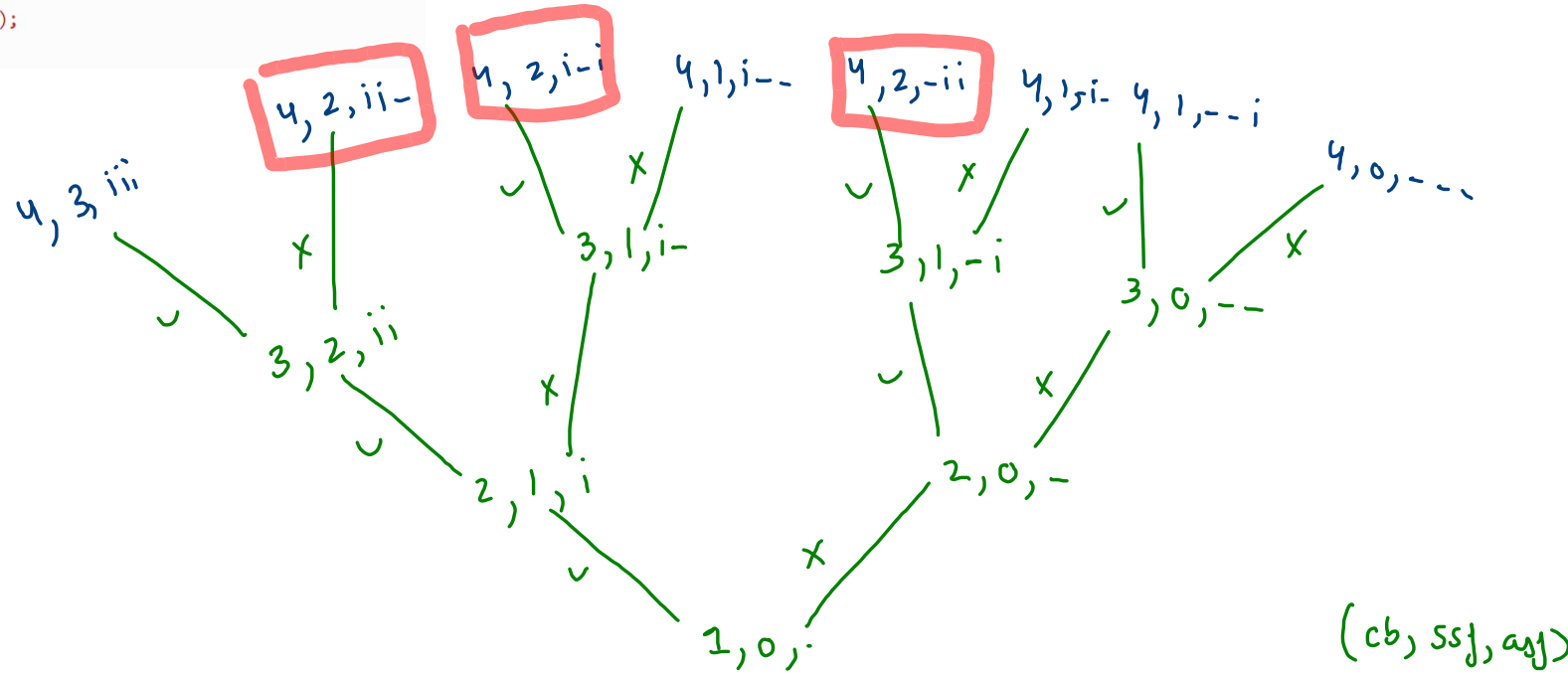
$$n = 3$$

$$r = 2$$

$$tb = 3$$

$$ts = 2$$

ii-  
 i-i  
 -ij

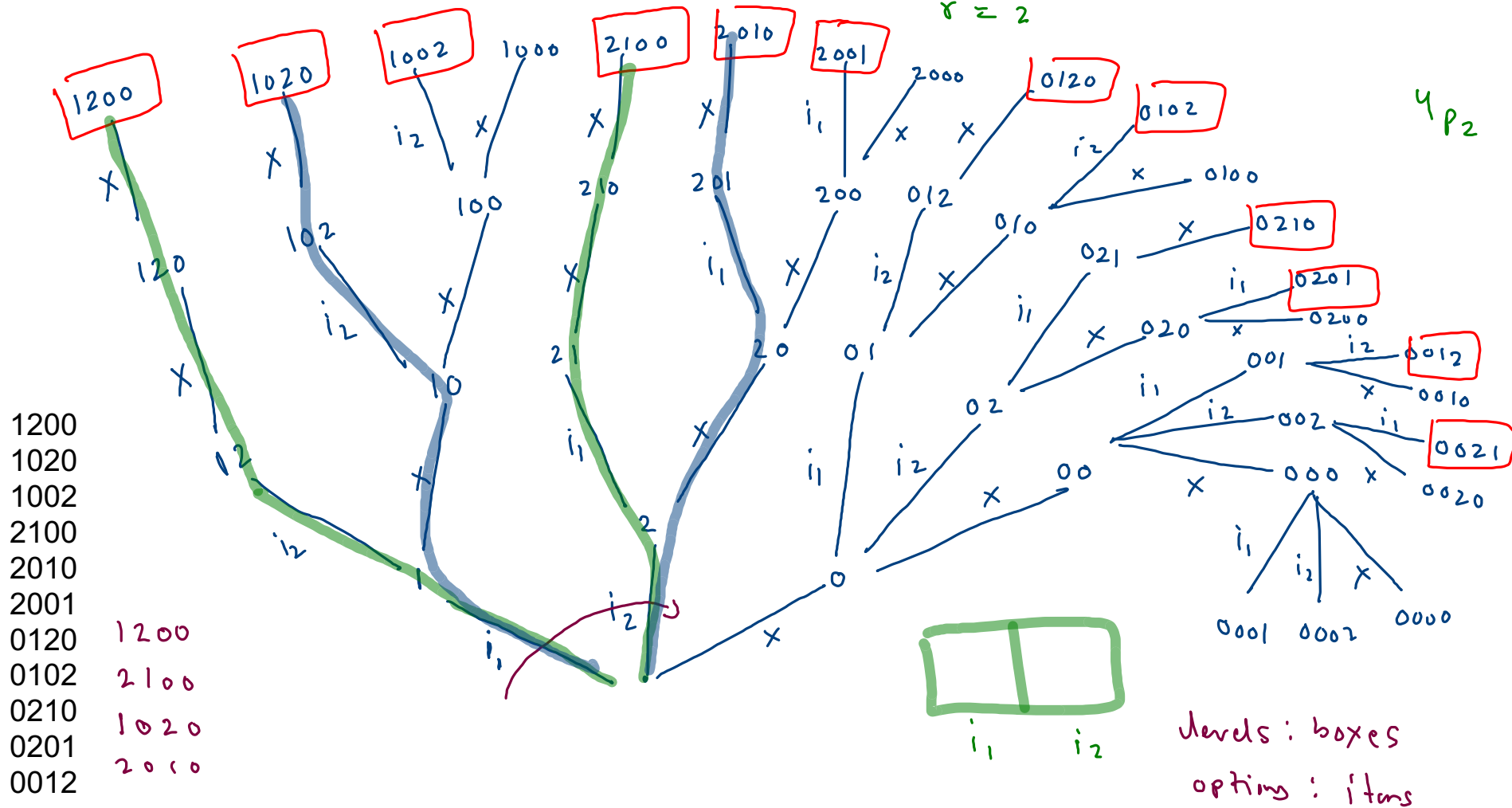


# Permutation - 2

$n = 4$

$r = 2$

$4P_2$



1200  
1020  
1002  
2100  
2010  
2001  
0120  
0102  
0210  
0201  
0012

1200  
2100  
1020  
2000



```

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

//box -> yes choice
for(int i=0; i < items.length;i++) {
    if(items[i] == 0) {
        items[i] = 1;
        permutations(cb+1,tb,items,ssf + 1,ts,asf + (i+1));
        items[i] = 0;
    }
}

//box -> no choice
permutations(cb+1,tb,items,ssf,ts,asf + 0);

```

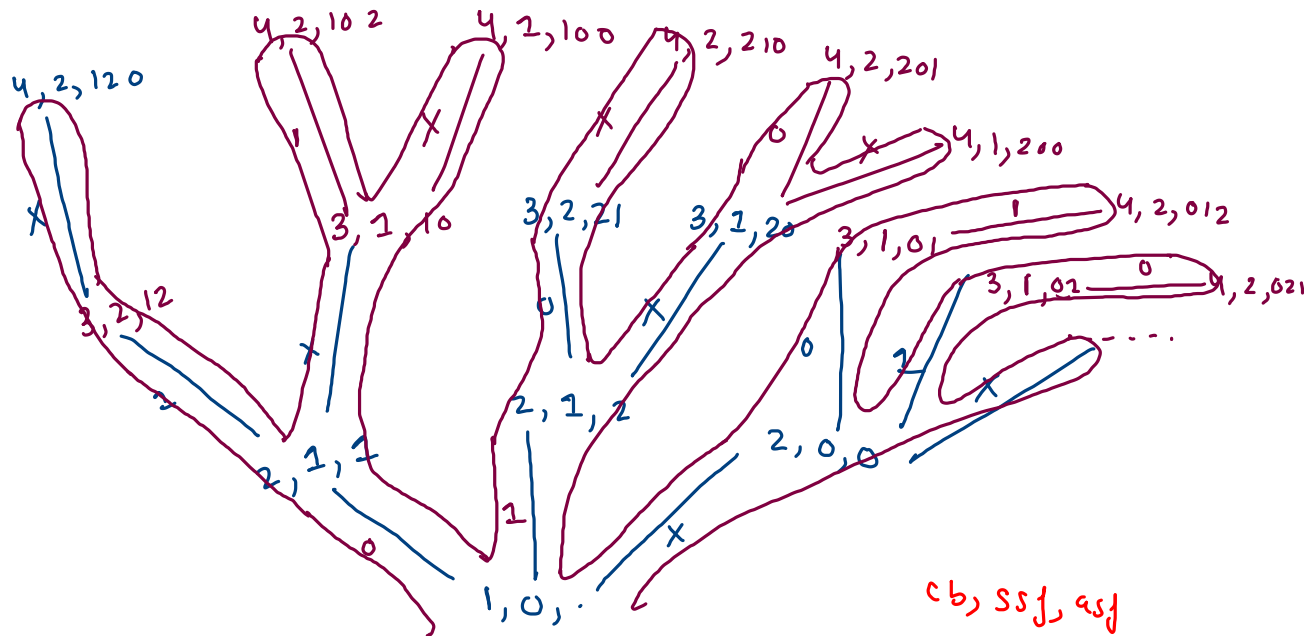
120  
 102  
 210  
 201  
 012  
 021

$n = 3$

$r = 2$

items
 

0	1



Combination 2

$$n = 4$$

$$r = 3$$



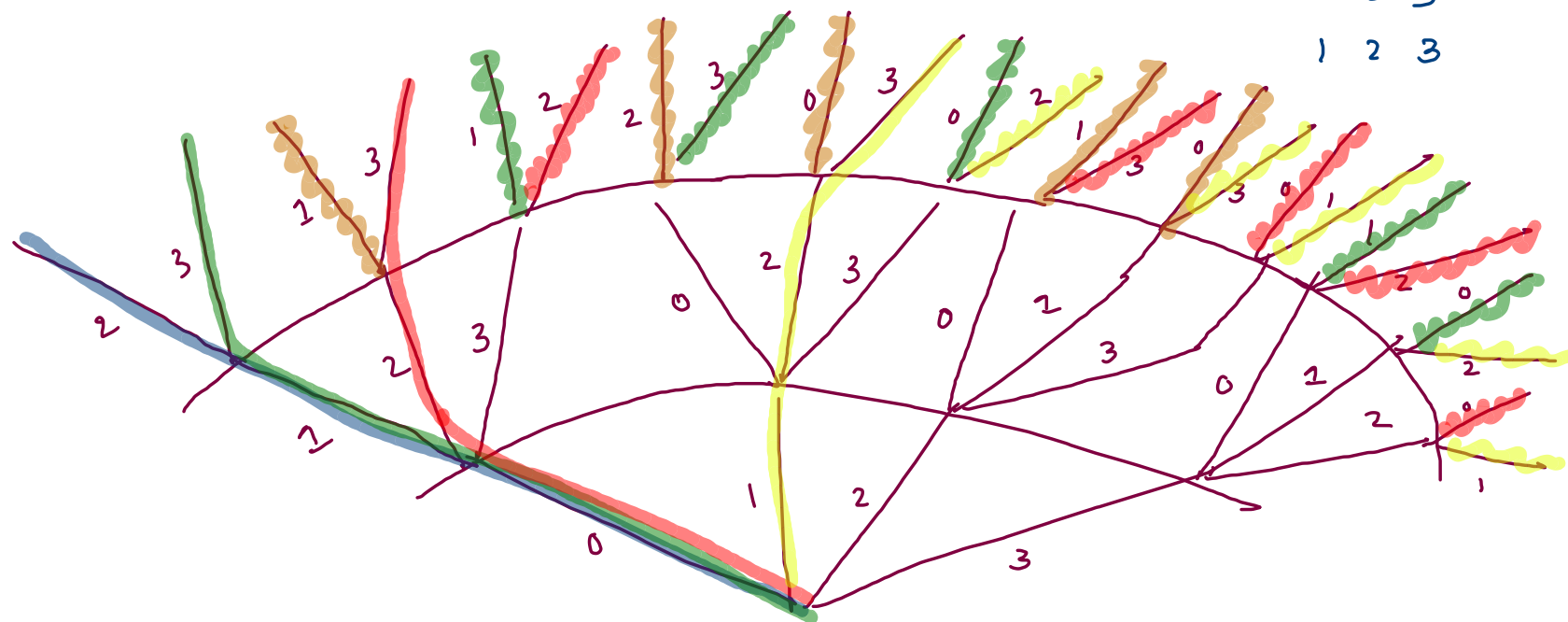
0 1 2

level  $\rightarrow$  items

0 1 3

0 2 3

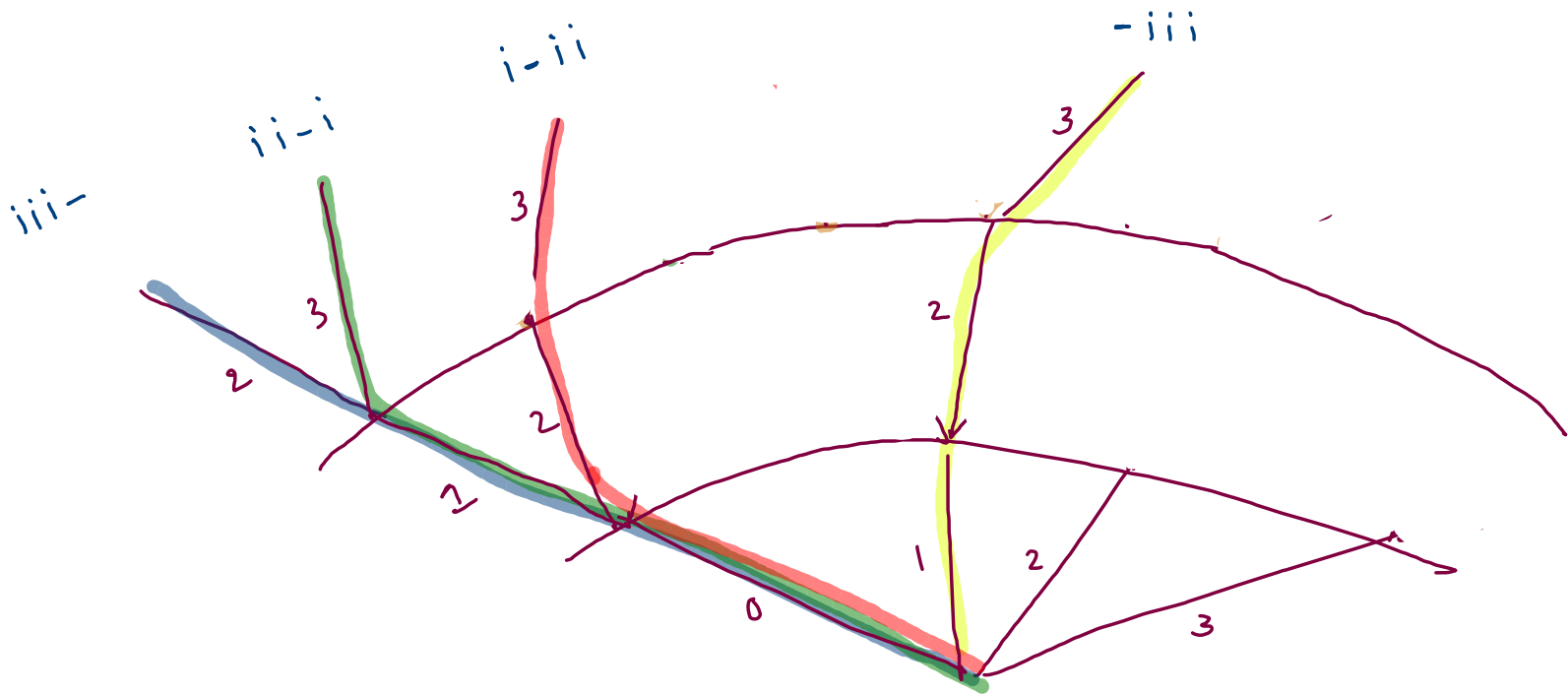
1 2 3



.

.

.





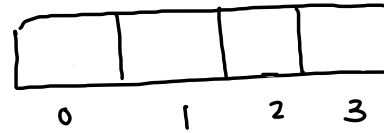
```

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;
    }

    //item choice -> which box to be select
    for(int b = lb+1; b < boxes.length;b++) {
        boxes[b] = 1;
        combinations(boxes,ci+1,ti,b);
        boxes[b] = 0;
    }
}

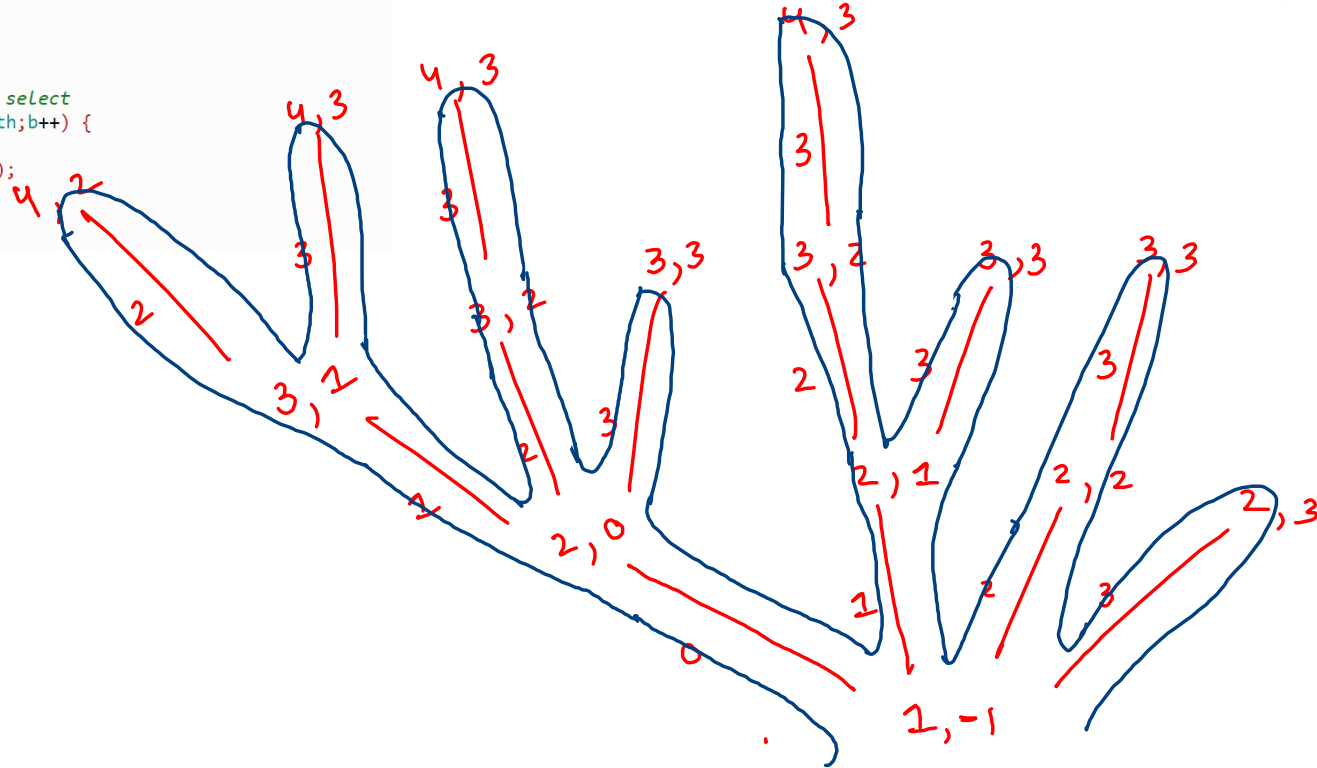
```

boxes



$$n = 4$$

$$r = 3$$



i i i -  
 i i - i  
 i - i i  
 - i i i

(ci, ti)