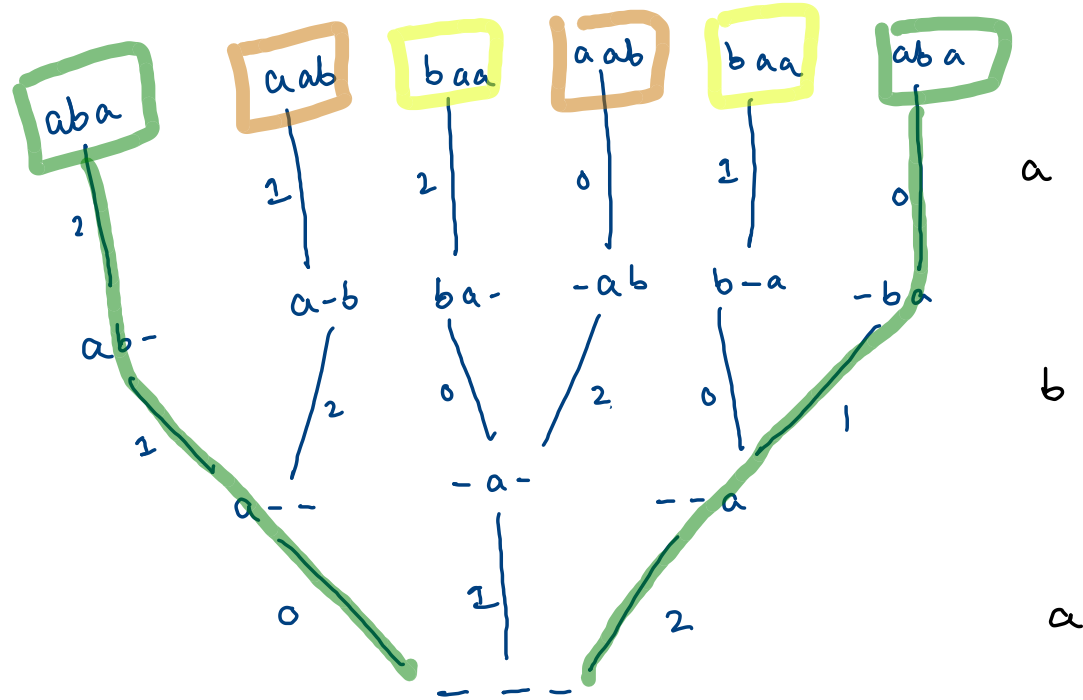
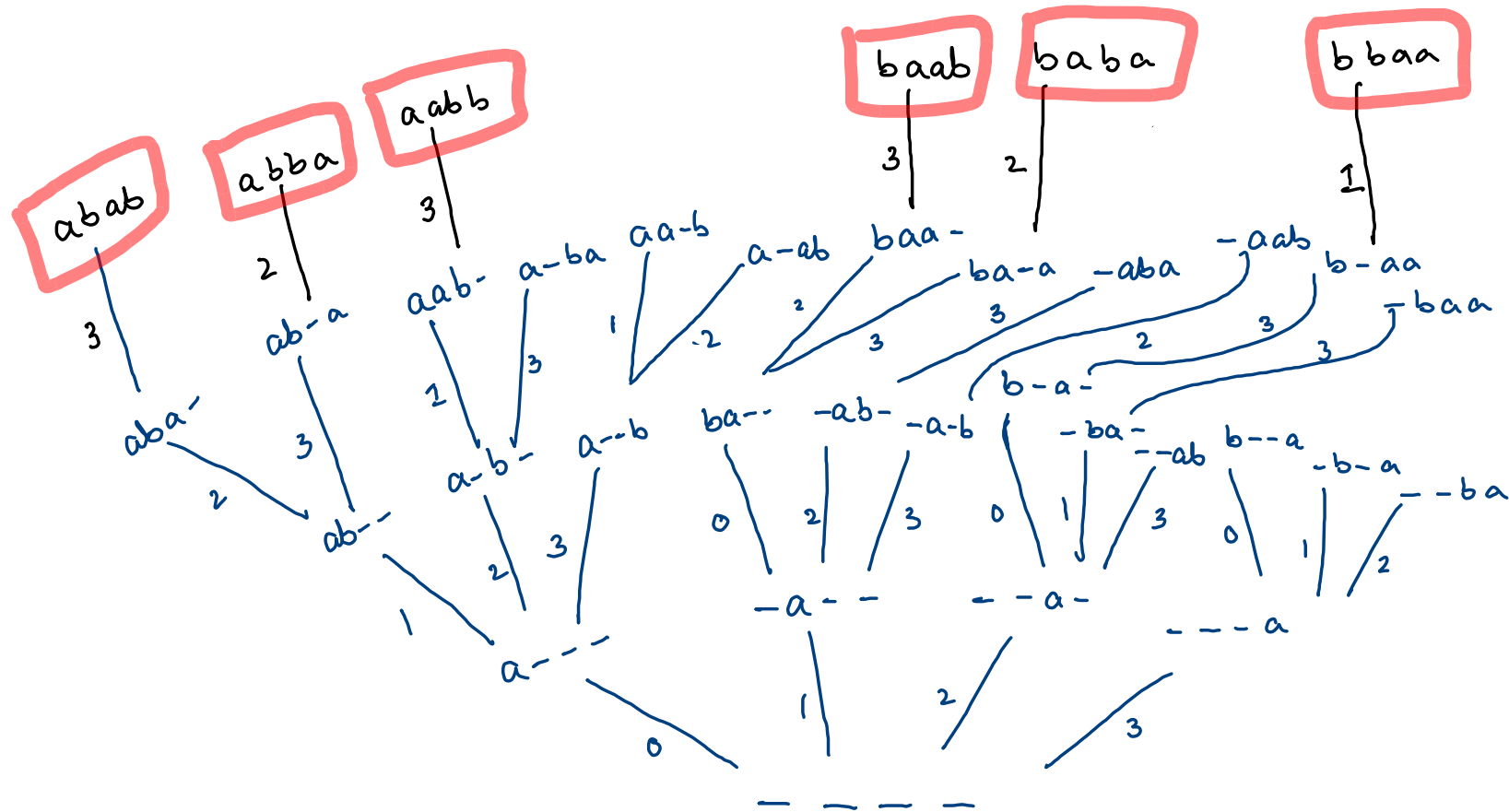


# Permutation - 2

Str = aba





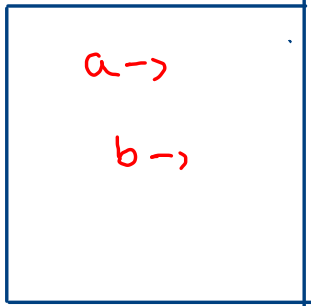
b

a

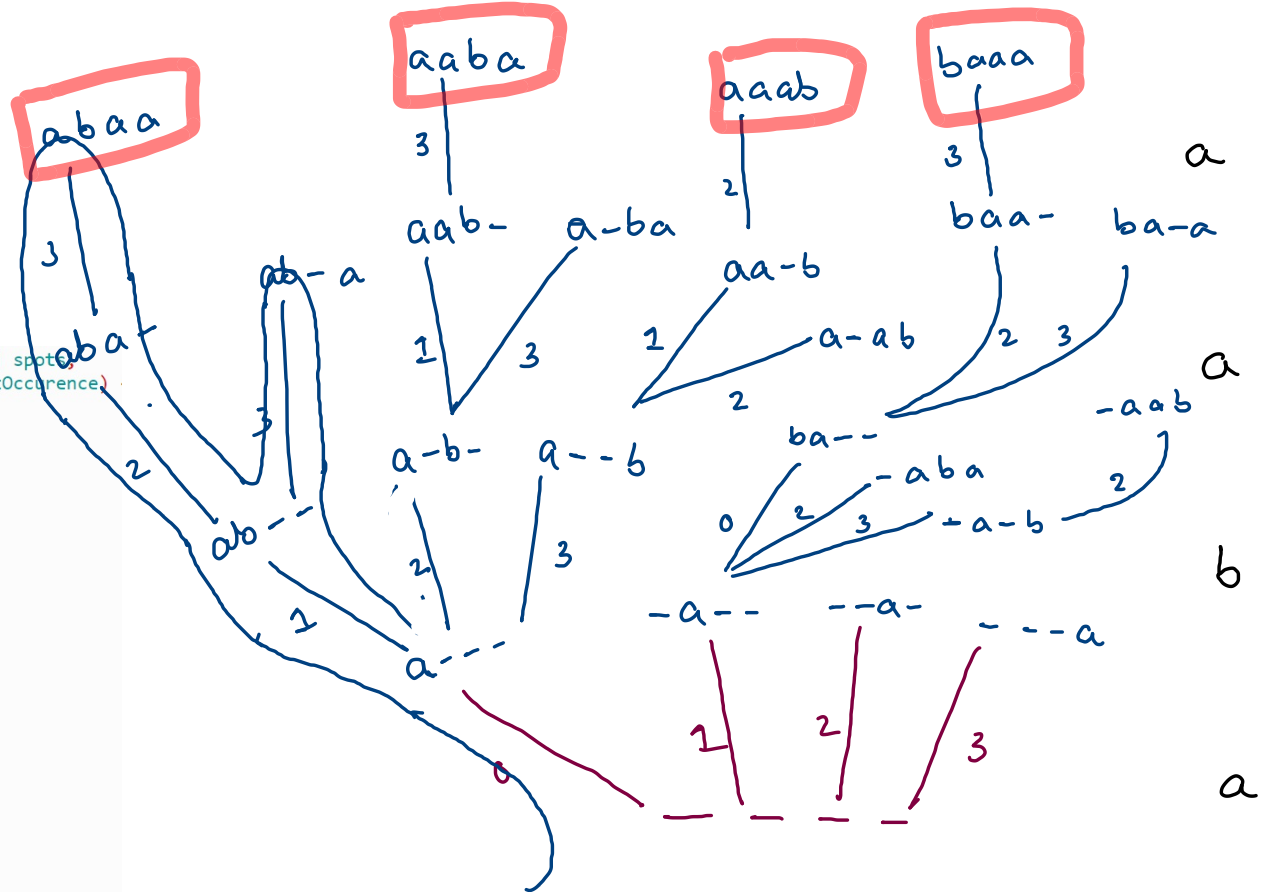
b

a

do



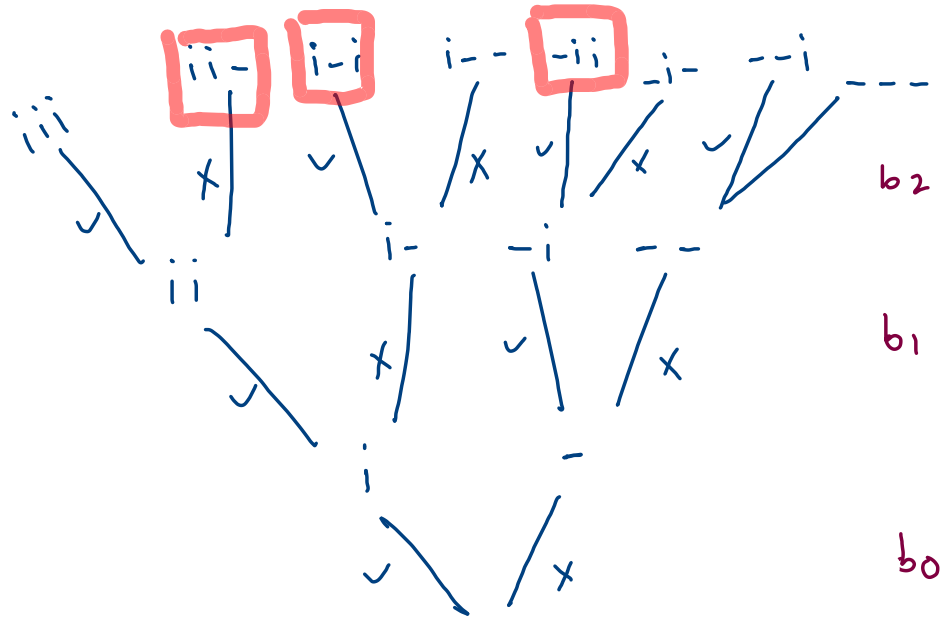
```
public static void generateWords(int idx, String str, Character[] spots,
    HashMap<Character, Integer> lastOccurence) {
    if(idx == str.length()) {
        for(int i=0; i < spots.length;i++) {
            System.out.print(spots[i]);
        }
        System.out.println();
        return;
    }
    char ch = str.charAt(idx);
    int loch = lastOccurence.get(ch);
    for(int i = loch + 1; i < spots.length;i++) {
        if(spots[i] == null) {
            spots[i] = ch;
            lastOccurence.put(ch,i);
            generateWords(idx+1,str,spots,lastOccurence);
            spots[i] = null;
            lastOccurence.put(ch,loch);
        }
    }
}
```



words k-selections 1

combination - 1

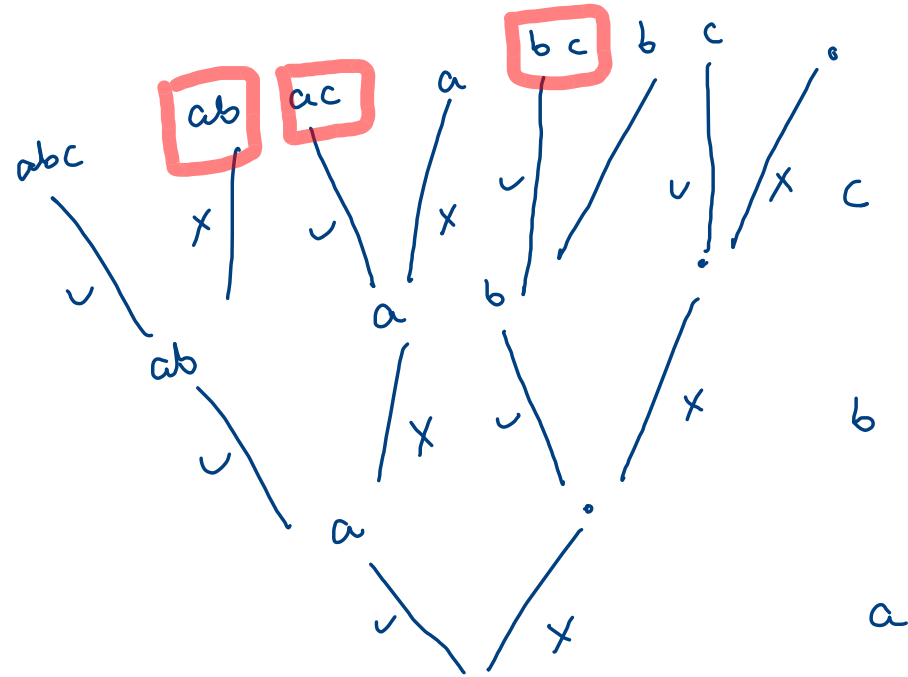
3 boxes  
2 items



Str = abcabc

ustr = abc

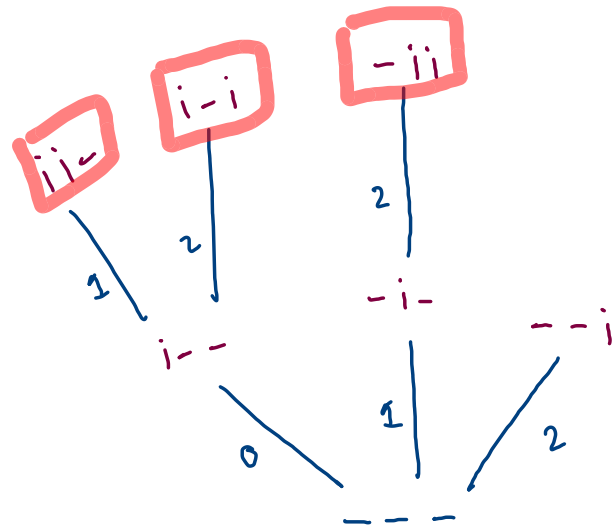
k = 2



words k-selection 2

combination-2

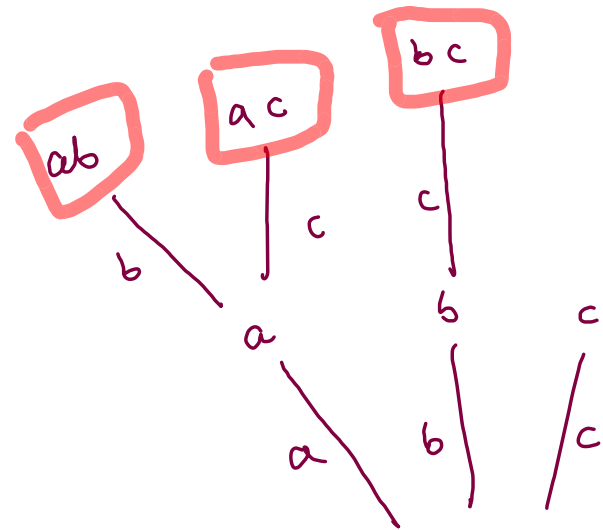
3 boxes  
2 items



str: abcabc

ustr: abc

k = 2



2

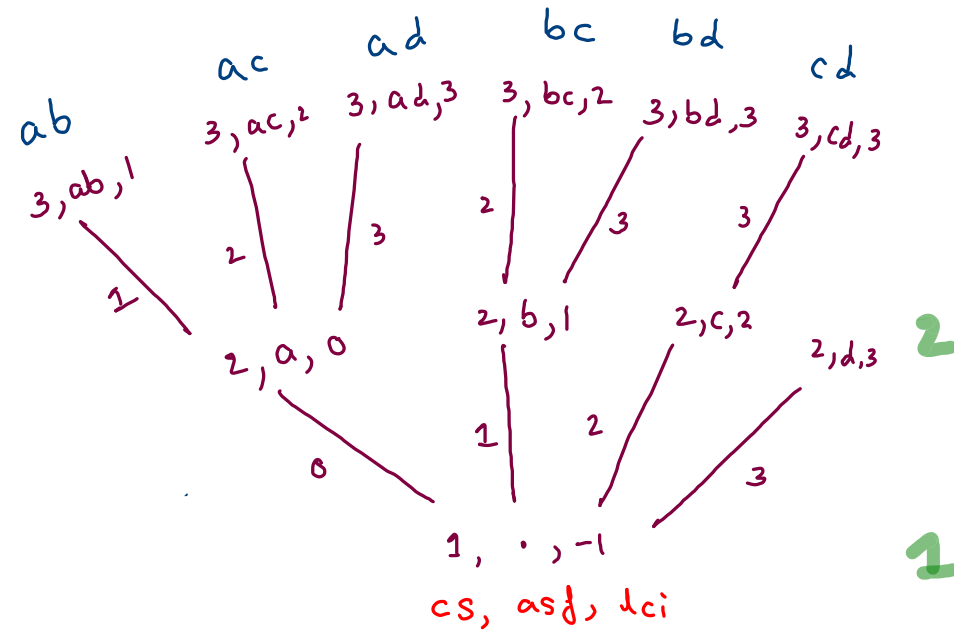
1

str = abc c d b a

ts = 2

ustr = a b c d

```
public static void words_k_selection2(int cs,int ts,String ustr,String asf,int lci) {  
    if(cs > ts) {  
        System.out.println(asf);  
        return;  
    }  
  
    for(int i= lci + 1; i < ustr.length();i++) {  
        char ch = ustr.charAt(i);  
        words_k_selection2(cs+1,ts,ustr,asf + ch,i);  
    }  
}
```

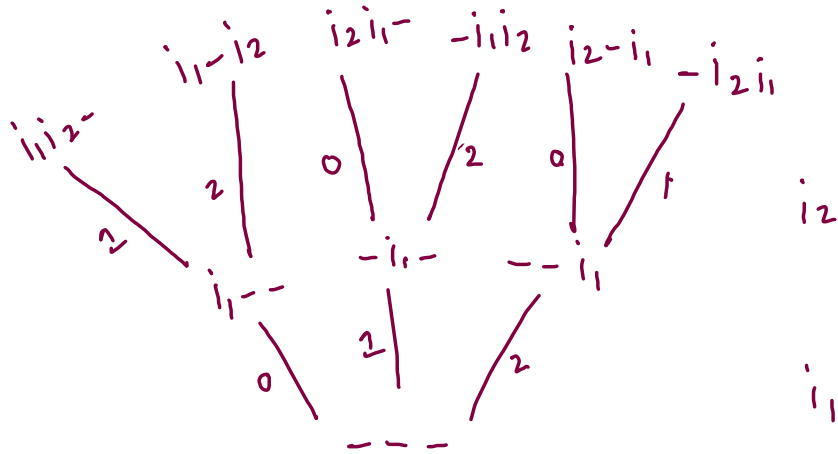


# words k arrangement -2

Permutations

3 boxes

2 items

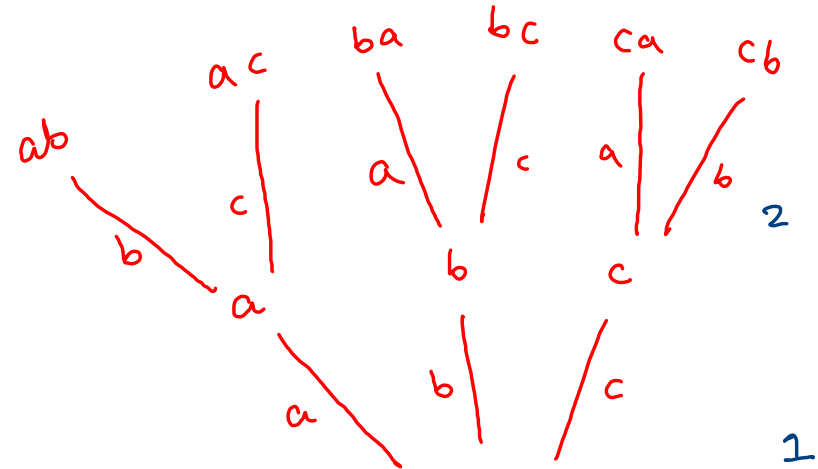


str: abcabc

ustr: abc

k = 2

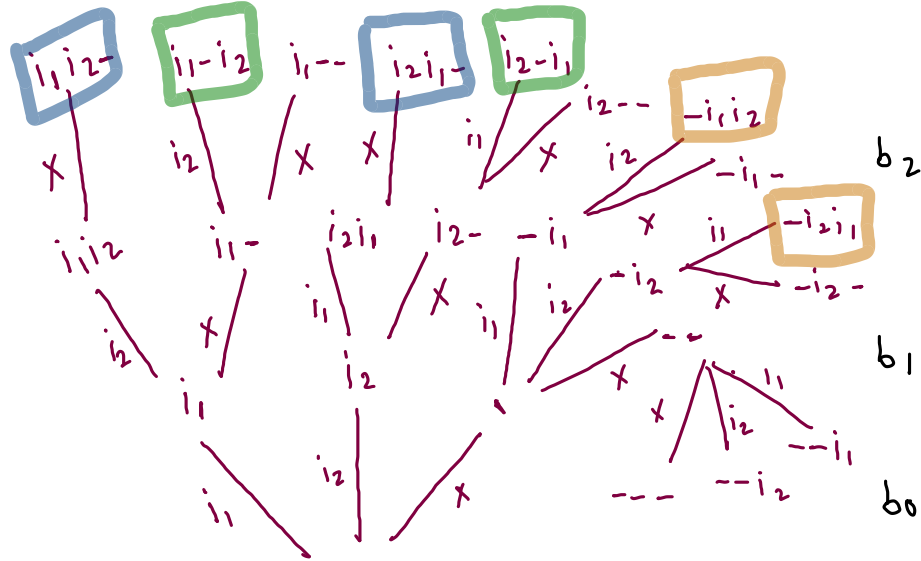
ab	ba
ac	ca
bc	cb



words k arrangement 1

permutations-2

3 boxes  
2 items



str = abcabc

ustr = abc

k = 2

