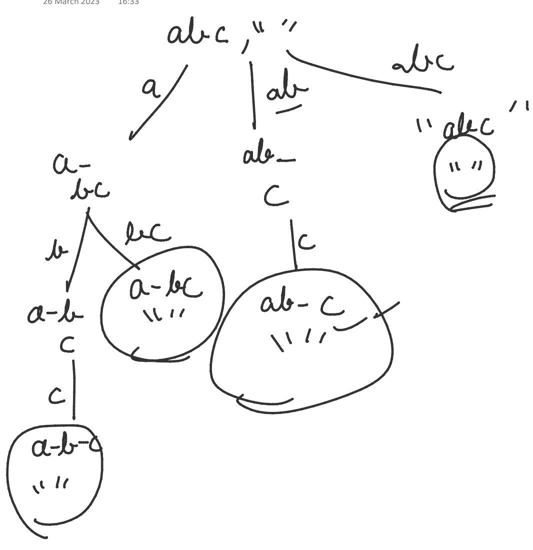
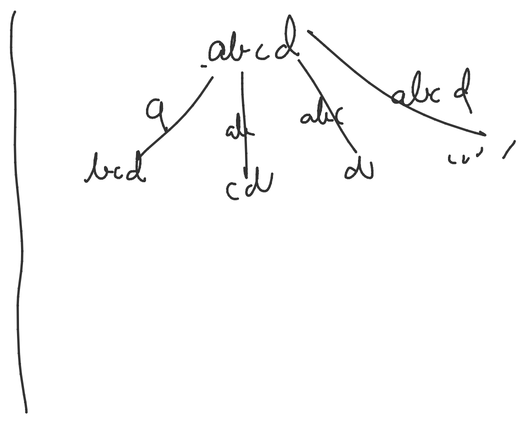
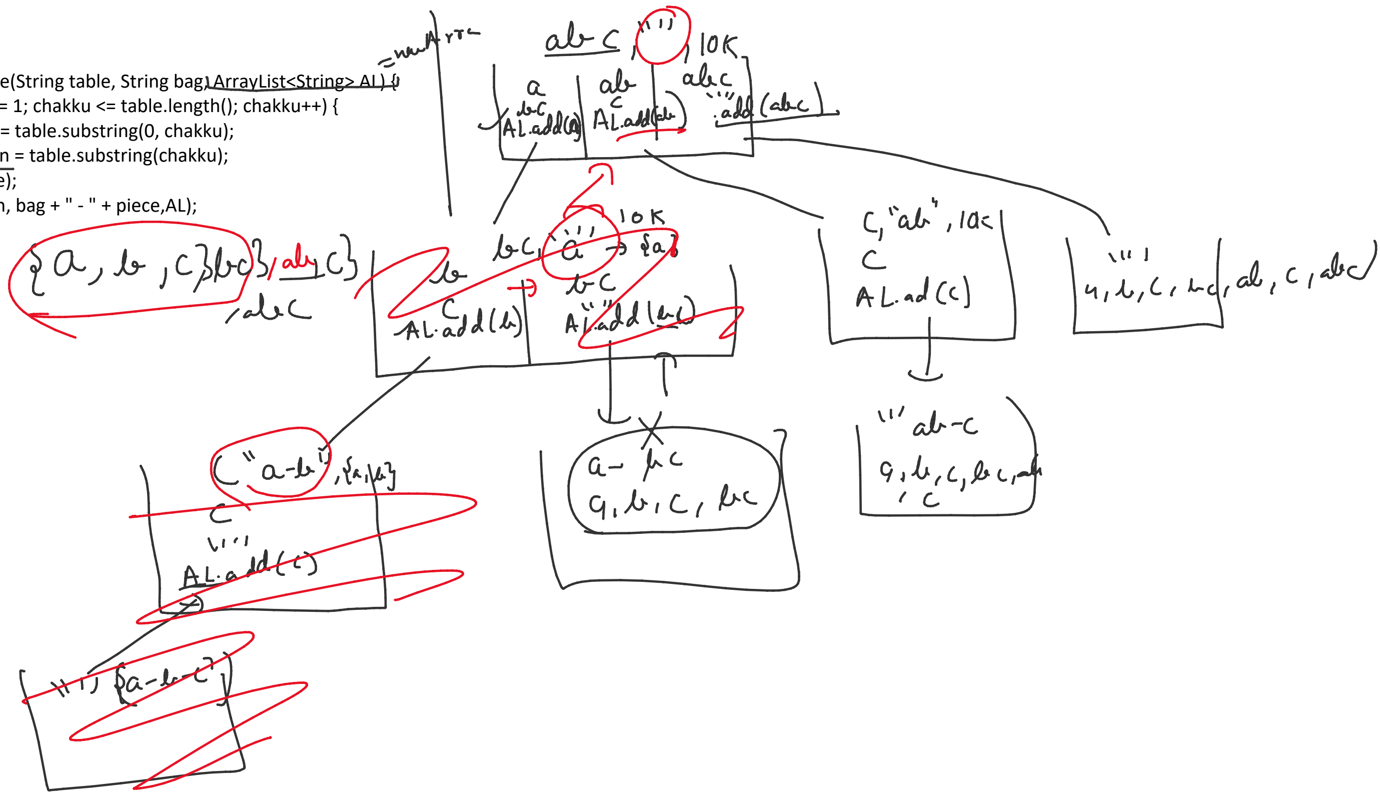
26 March 2023 16:33

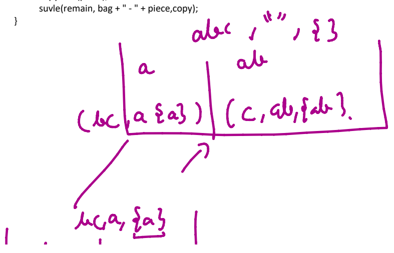


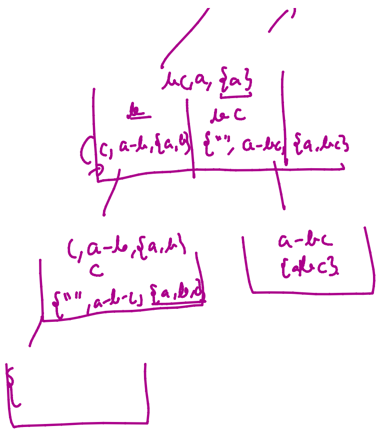


```
public static void suvle(String table, String bag) ArrayList<String> AL) {
        for (int chakku = 1; chakku <= table.length(); chakku++) {
            String piece = table.substring(0, chakku);
            String remain = table.substring(chakku);
            AL.add(piece);
            suvle(remain, bag + " - " + piece,AL);
        }
    }
```

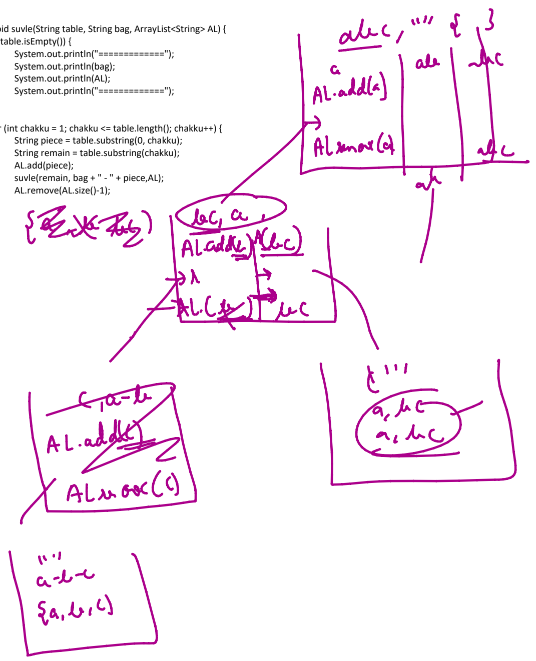


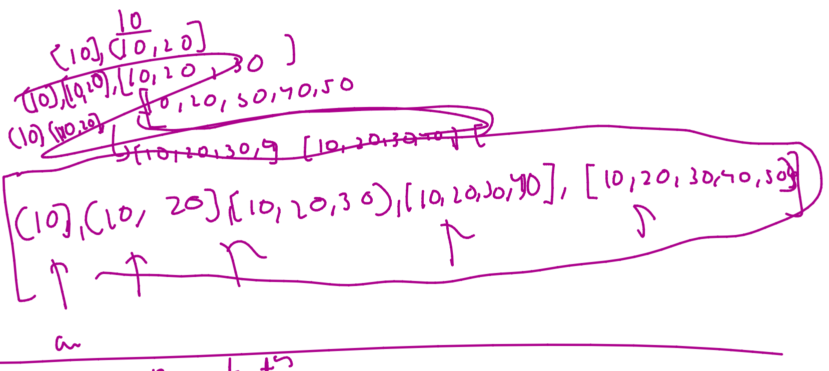
```
for (int chakku = 1; chakku <= table.length(); chakku++) {
                String piece = table.substring(0, chakku);
                String remain = table.substring(chakku);
//                      AL.add(piece);
                ArrayList<String> copy = new ArrayList<>(AL);
                copy.add(piece);
                suvle(remain, bag + " - " + piece,copy);
        }
```





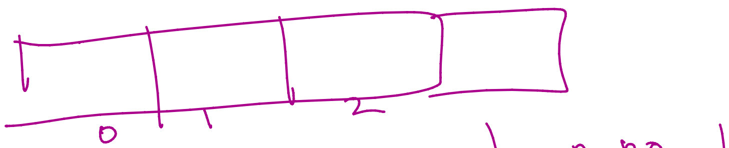
```
public static void suvle(String table, String bag, ArrayList<String> AL) {
            if (table.isEmpty()) {
                  System.out.println("=============");
                  System.out.println(bag);
                  System.out.println(AL);
                  System.out.println("=============");

            }
            for (int chakku = 1; chakku <= table.length(); chakku++) {
                  String piece = table.substring(0, chakku);
                  String remain = table.substring(chakku);
                  AL.add(piece);
                  suvle(remain, bag + " - " + piece,AL);
                  AL.remove(AL.size()-1);
            }
      }
```





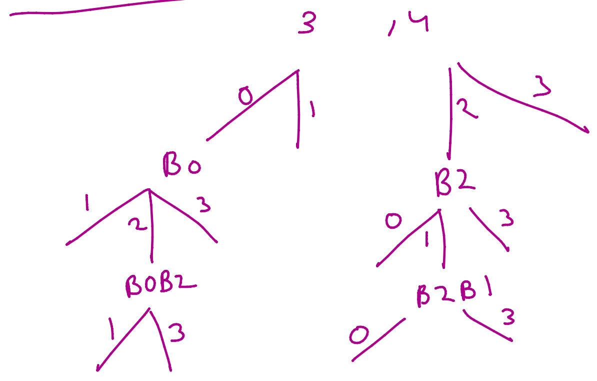
Permutatⁿ

2 → 4

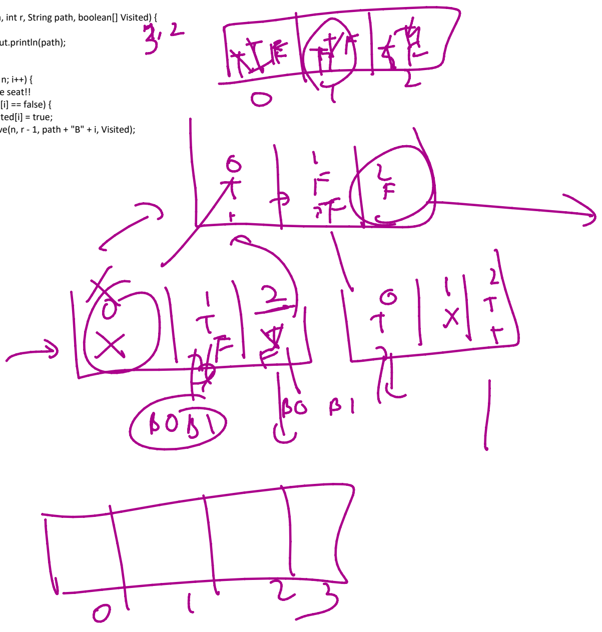


B0 B1	B1 B0	B2 B0	B3 B0
B0 B2	B1 B2	B2 B1	B3 B1
	B1 B3	B2 B3	B3 B2

B0 B1 | B1 B2 | B2 B1 | B3 B1
B0 B2 | B1 B3 | B2 B3 | B3 B2
B0 B3



```
public static void solve(int n, int r, String path, boolean[] Visited) {
                if (r == 0) {
                        System.out.println(path);
                        return;
                }
                for (int i = 0; i < n; i++) {
//                              i the seat!!
                        if (Visited[i] == false) {
                                Visited[i] = true;
                                solve(n, r - 1, path + "B" + i, Visited);
                        }
                }
        }
```



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