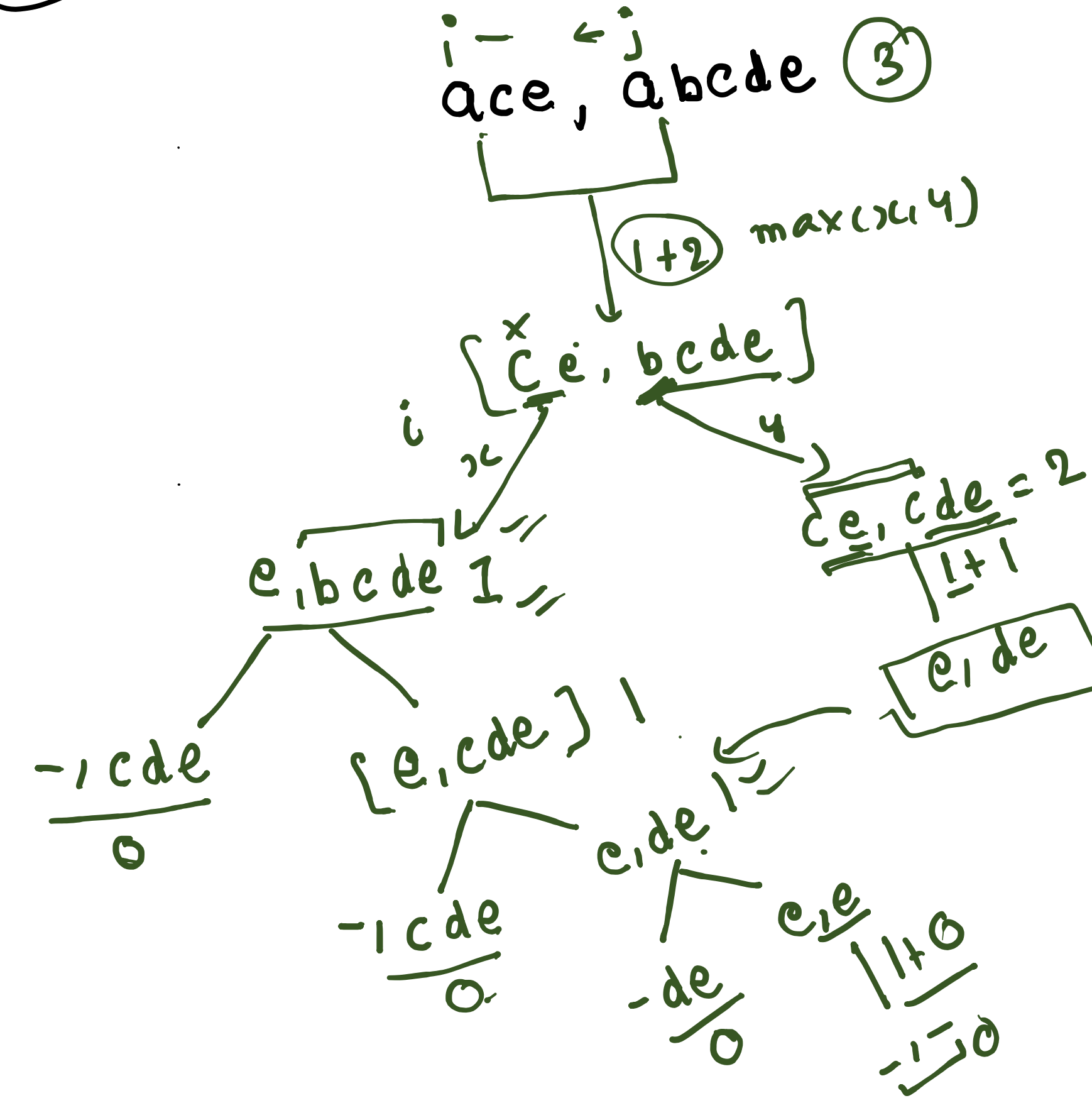
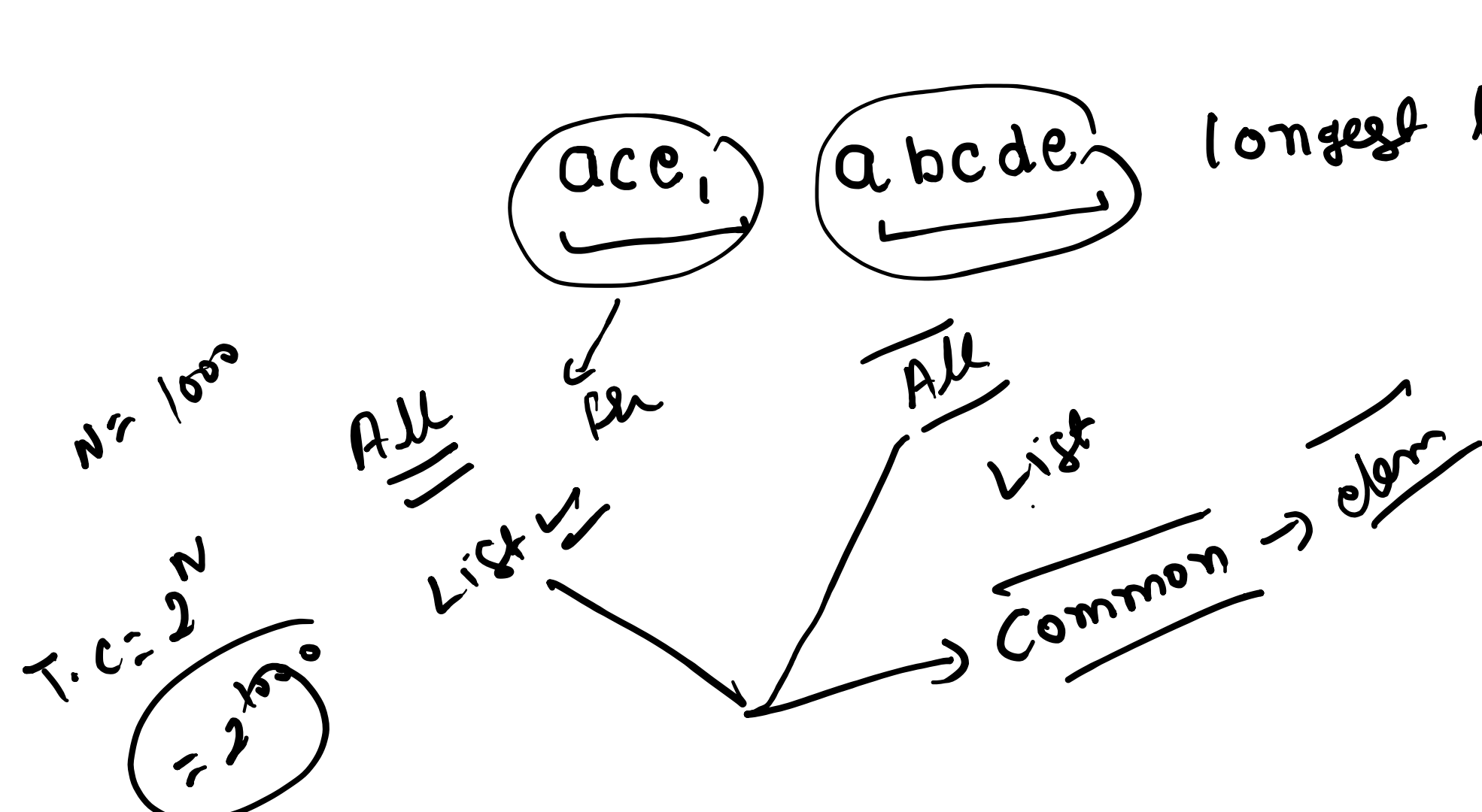
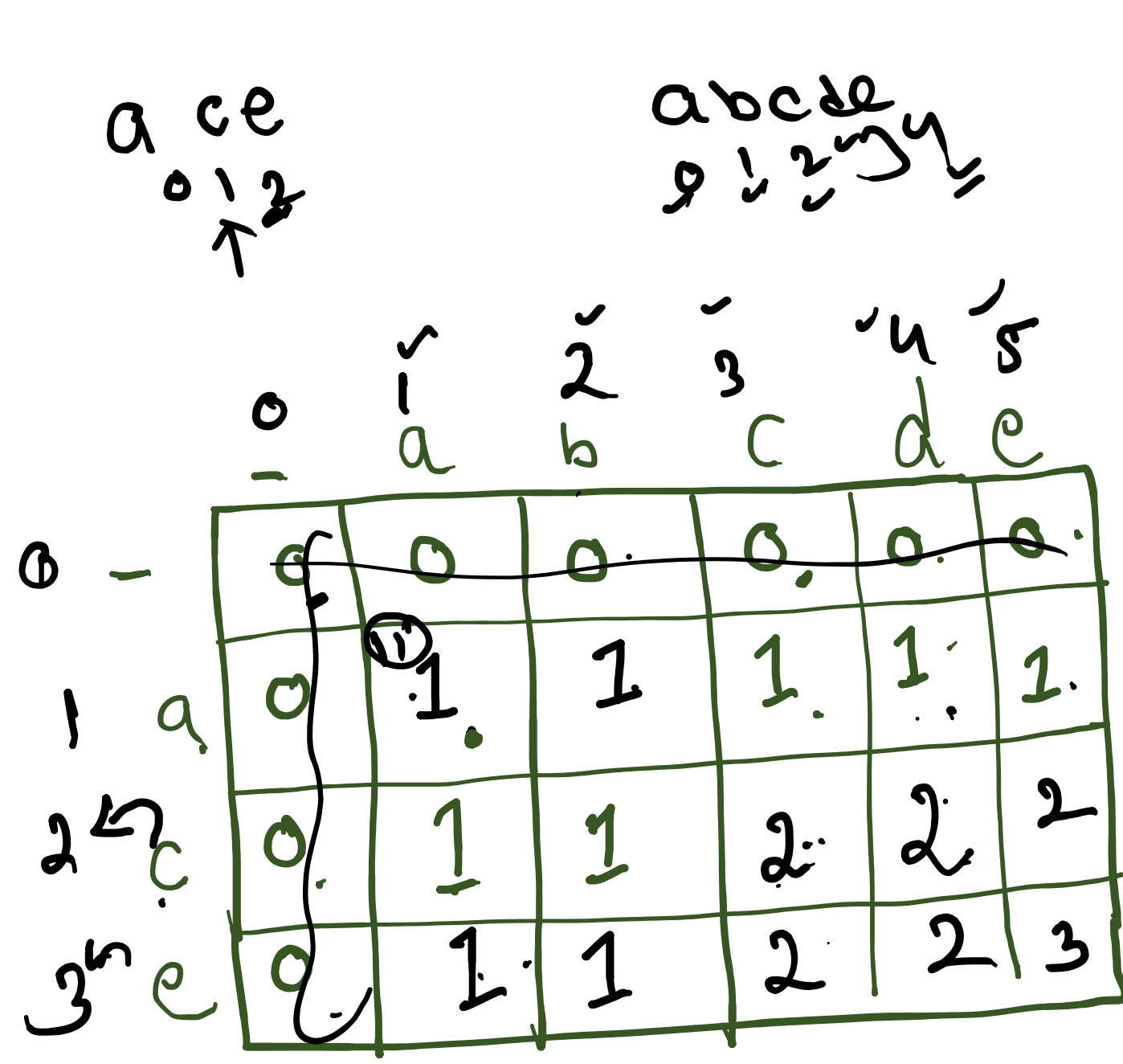
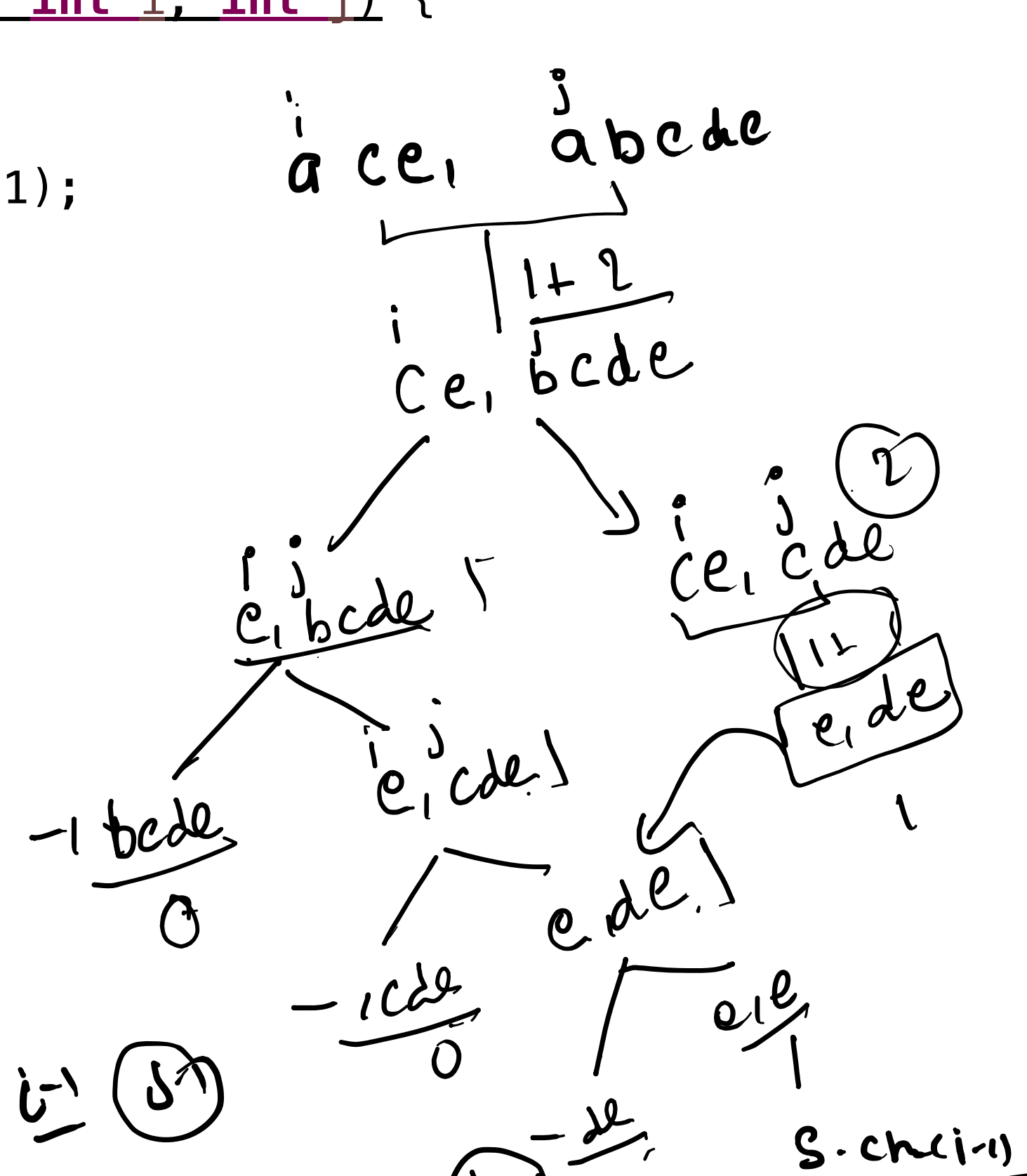


longest length common



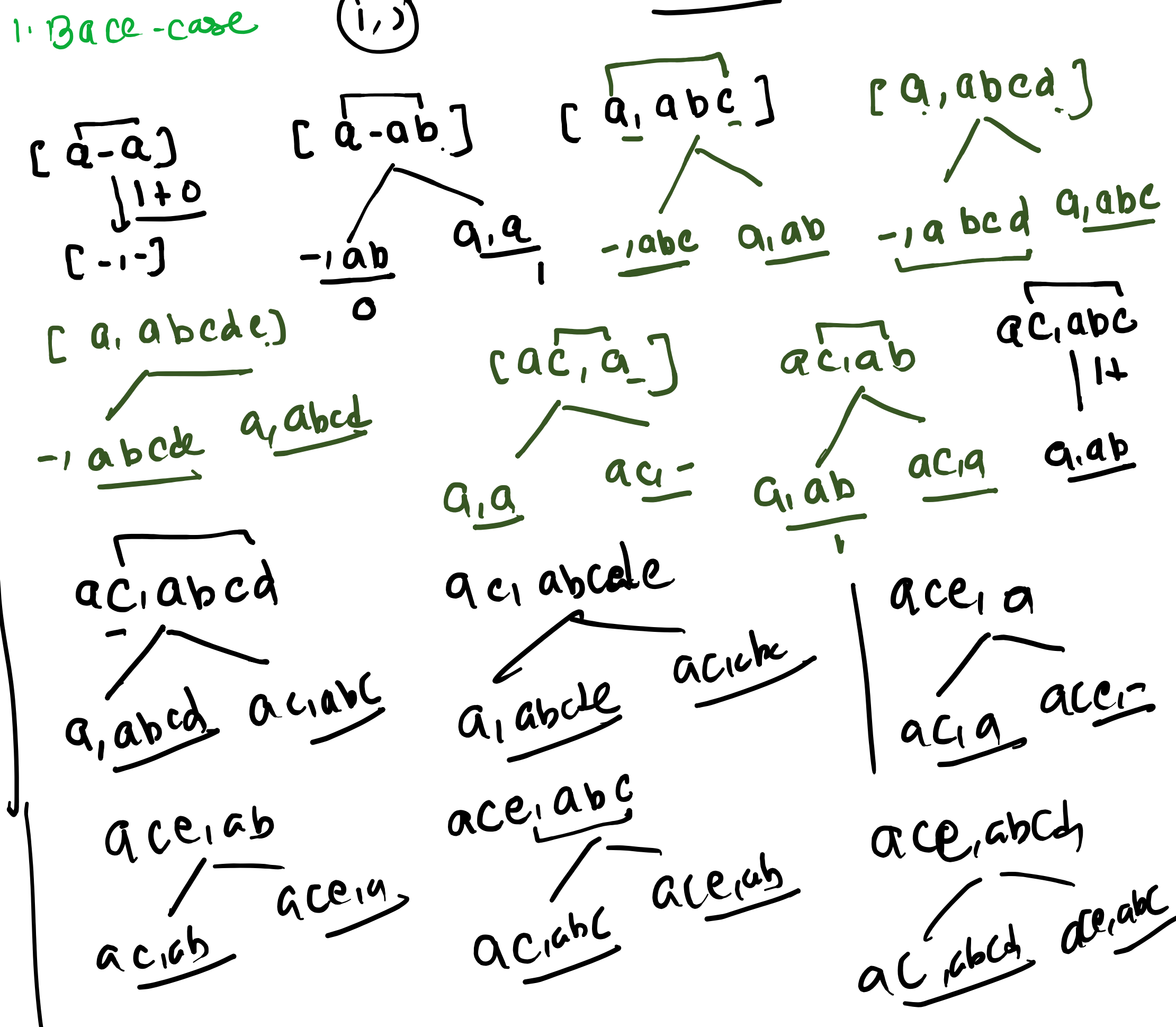
```
public static int LCS(String s, String t, int i, int j) {
```

```
public static int LCS(String s, String t, int i, int j) {
    if (ans == 0) {
        if (s.charAt(i) == t.charAt(j)) {
            ans = 1 + LCS(s, t, i + 1, j + 1);
        } else {
            int f1 = LCS(s, t, i + 1, j);
            int s1 = LCS(s, t, i, j + 1);
            ans = Math.max(f1, s1);
        }
    }
}
```



miss rate / $\text{decis}(i) = \left\{ \begin{array}{l} \text{decis}(i) \\ \text{decis}(i-1) \end{array} \right\}$

malan $\rightarrow d p(c i) c j = 1 + d p(c i - 1) c j$
aceibcde
1 1

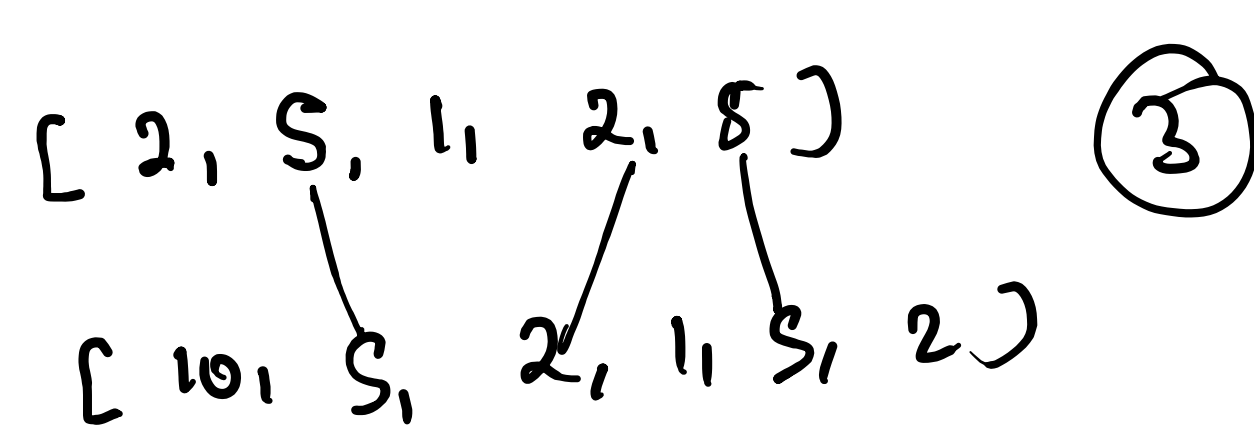


L.C.S ~~arr~~

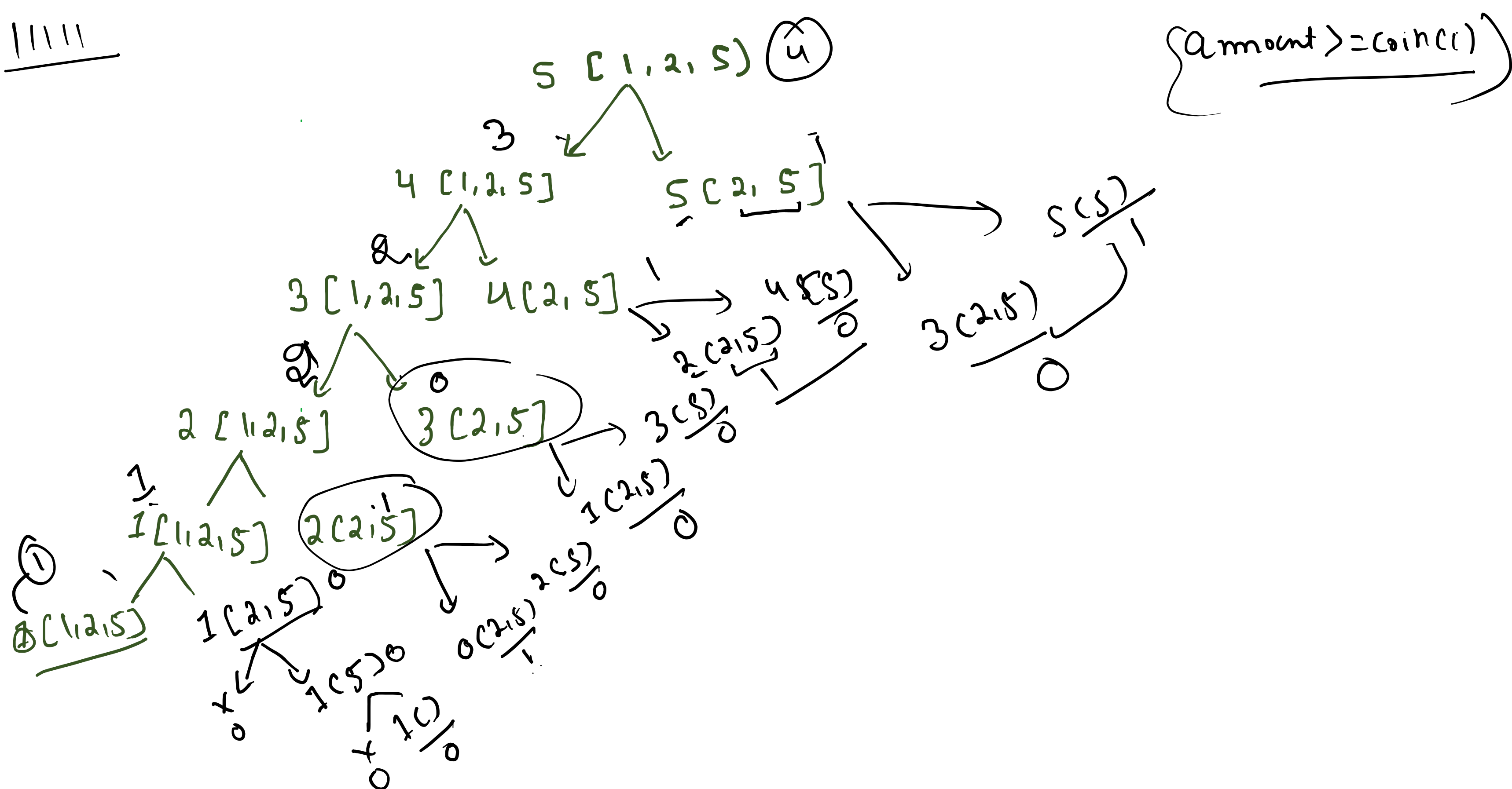
a c e
 a b c d e

```
nums1 = [2,5,1,2,5]
```

```
nums1 = [2,5,1,2,5]
nums2 = [10,5,2,1,5,2]
```



|||||

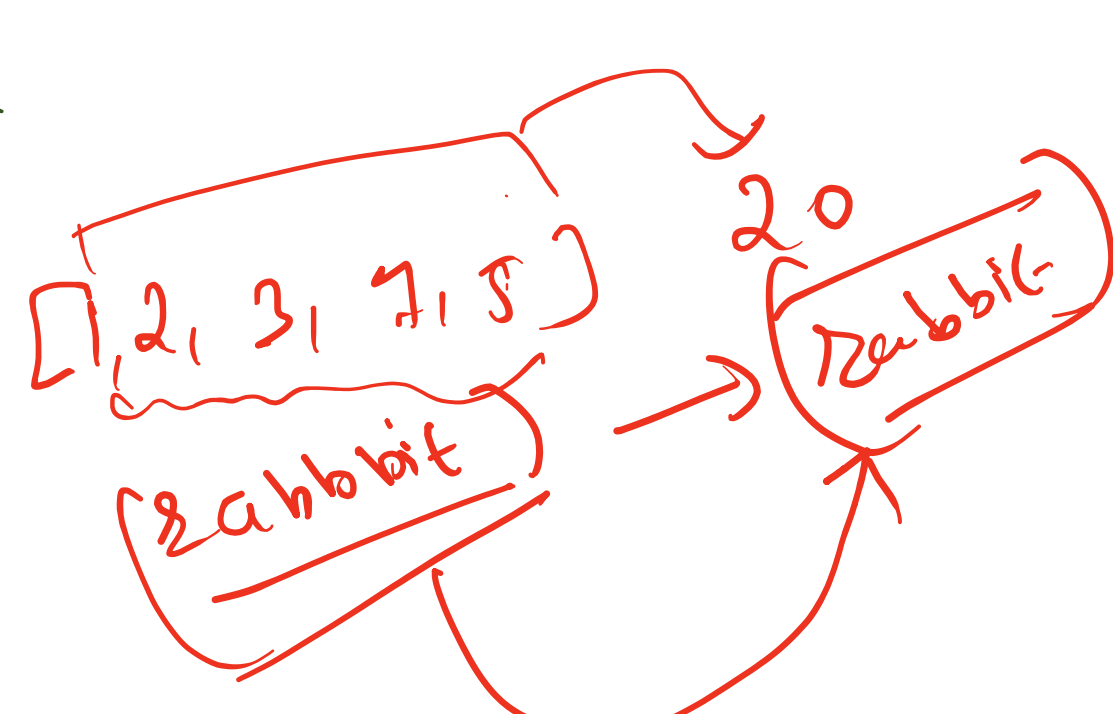
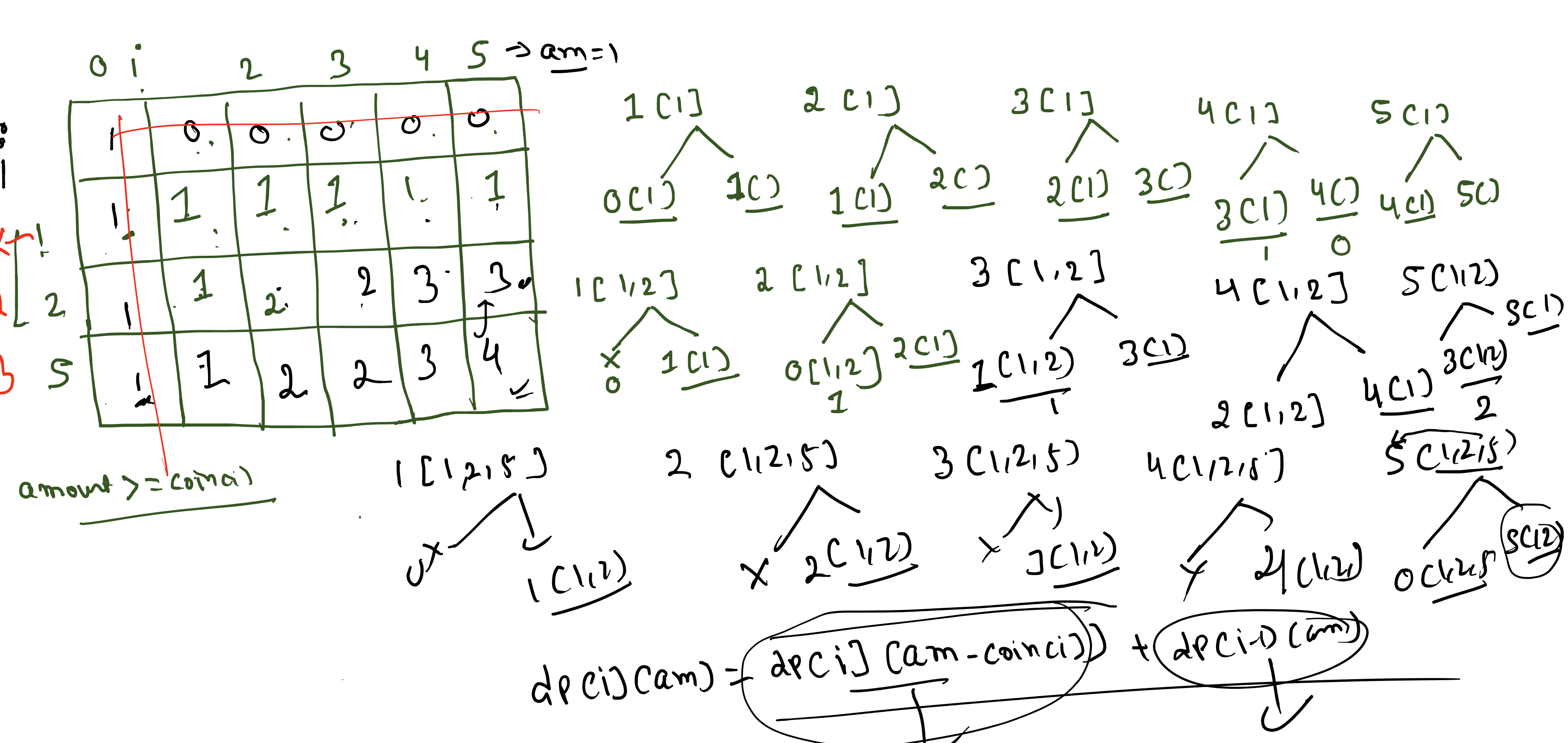
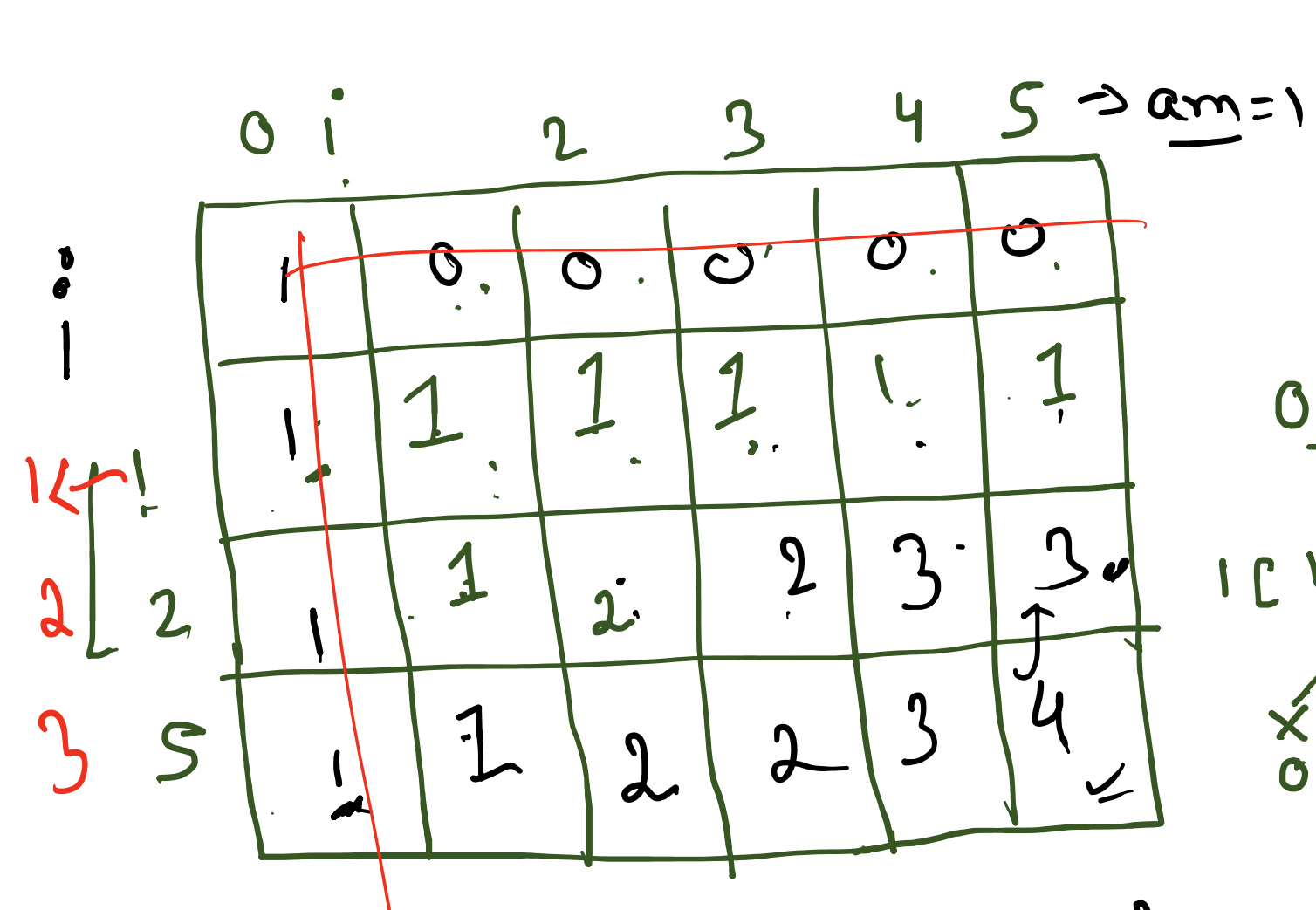
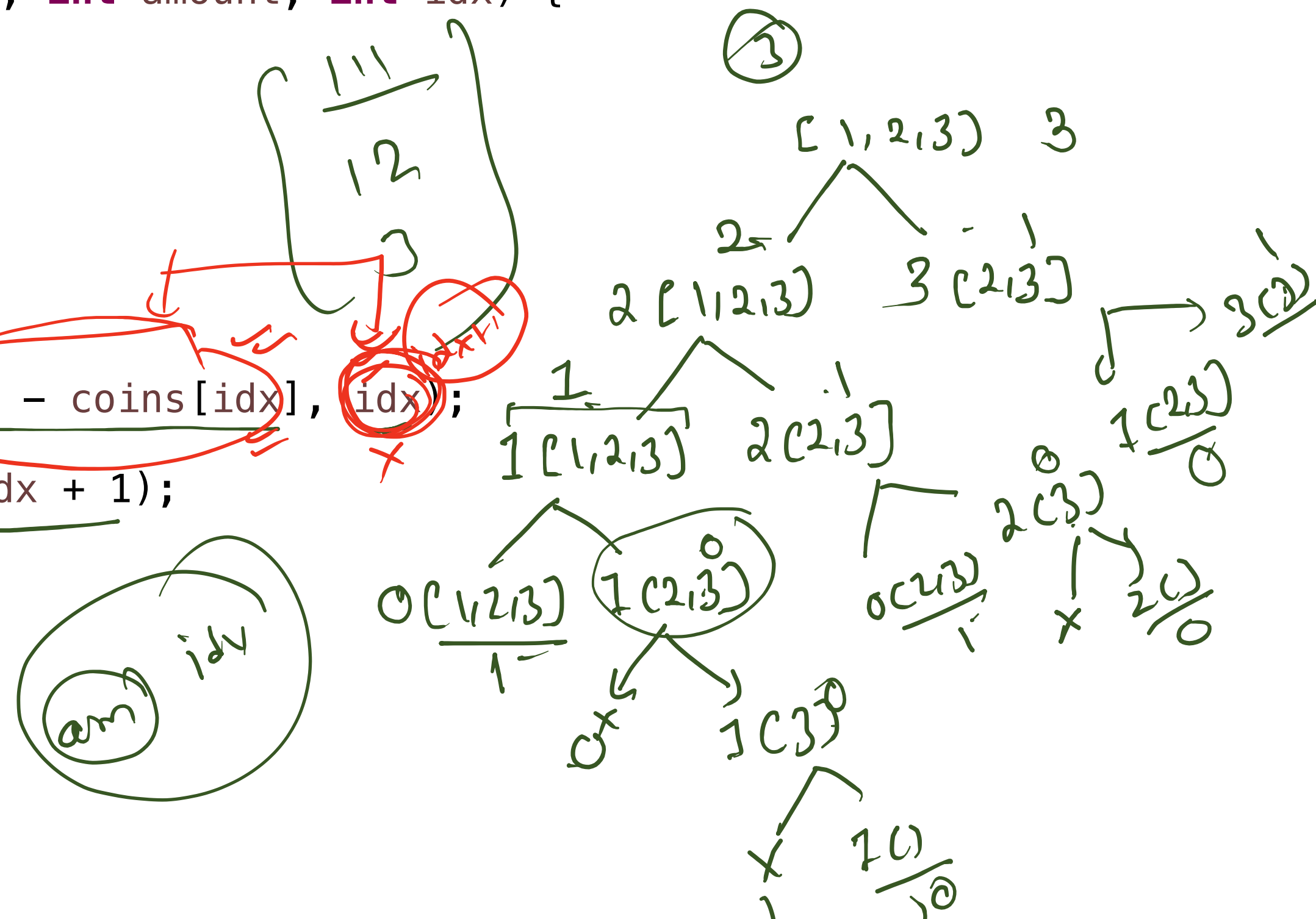


```
public static int Coin Change(int[] coins, int amount, int idx) {
```

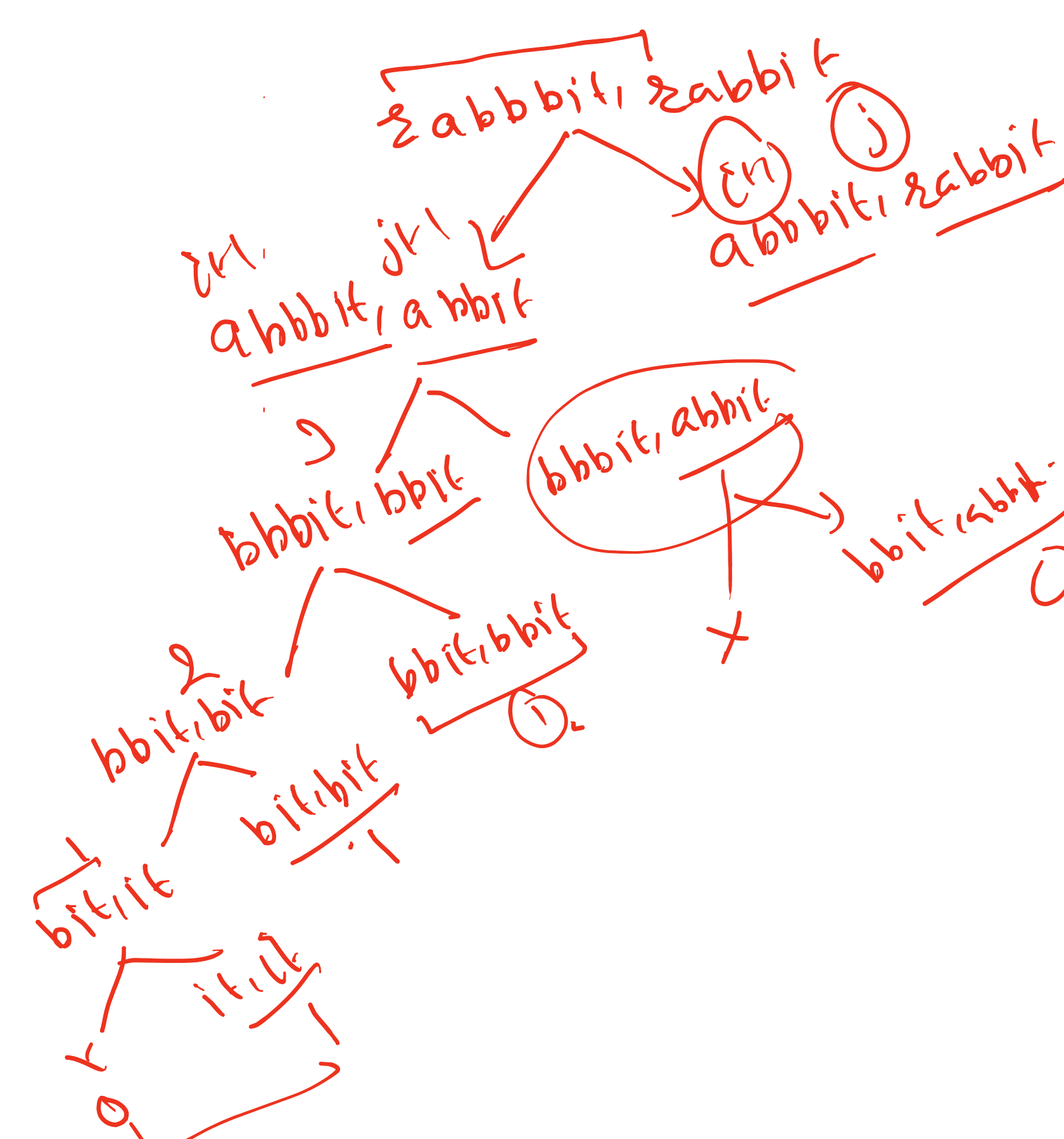
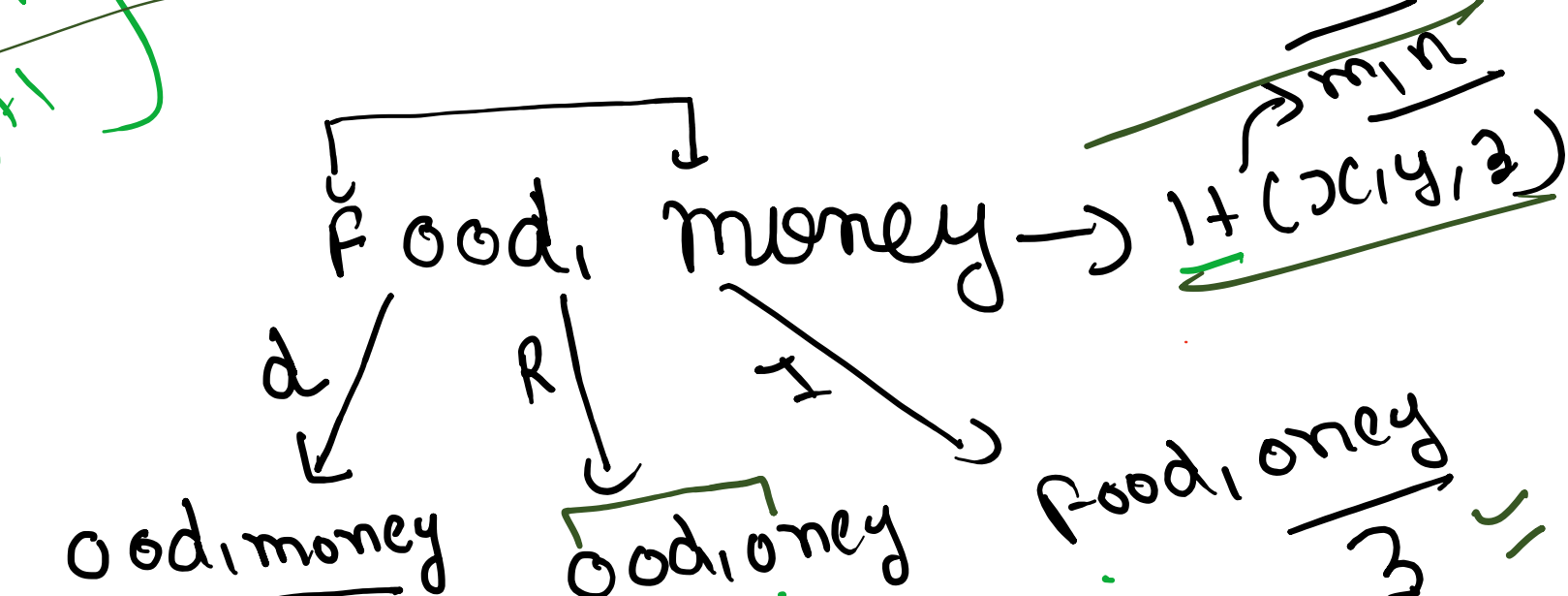
```

public static int Coin_Change(int[] coins, int amount, int idx) {
    if (amount == 0) {
        return 1;
    }
    if (idx == coins.length) {
        return 0;
    }
    int inc = 0, exc = 0;
    if (amount >= coins[idx]) {
        inc = Coin_Change(coins, amount - coins[idx], idx);
    }
    exc = Coin_Change(coins, amount, idx + 1);
    return inc + exc;
}

```



Quesada


$$\begin{bmatrix} d = 1 + j \\ R = 1 + j + 1 \\ I = 2, 5 + 1 \end{bmatrix}$$


mood, money
mood, money

1. 4. 21

edicy
edicy

$\sqrt{n \ln n}$

