

Class - 48

* Permutations

next-Permutation

S = "ab c" \Rightarrow

" ab c"

Recursion

① Using extra space

String Out = ""

~~Print Permutation (vector<int> nums, vector<int> output, int freq[], int id)~~

2) ~~int n = nums.size();~~

~~if (output.size() == n)~~

~~{ cout << output << endl;~~

~~}~~

~~for (int i = 0; i < nums.size(); i++)~~

~~if (!freq[i])~~

~~output.push_back(nums[i]);~~

~~freq[i] = 1;~~

~~Print Permutation (nums, output, freq);~~

~~freq[i] = 0;~~

~~cout << " " << back();~~

```

main()
{
    vector<int> nums;
    vector<int> output;
    vector<vector<int>> freq;
    int freq[100000000];
    for (int i = 0; i < freq.size(); i++)
        freq[i] = 0;
}

```

freq[i] = 0;

0, 0, 0, 4

0, 2, 3

freq[si] = 0;

Output · p₀ p₋ back()



1, 2, 3

3, 2, 1, 4

4

2, 1, 3, 4

Output [-]

[-]

[0, 0, 0]

freq [3] = [-]

Output [+] freq [1, 0, 0]

[1, 2, 3, [1, 1, 0]]

(1, 2, 3), [1, 1, 1]

[1, 3, -], [1, 0, 1]

[1, 3, 2], [1, 1, 1]

[2, - -], [0, 1, 0]

[2, 1, -], [1, 1, 0] (2, 3, -), [0, 1, 1]

[2, 3, 1], [1, 1, 1]

1, 2, 3, 4

Output.size() - 1

[3, 0, -] { 0, 0, 1 }

{ 3, 1, - } { 1, 0, 0 }

{ 3, 2, - } { 0, 1, 1 }

{ 3, 1, 2 }

{ 3, 2, 1 } { 1, 1, 1 }

Permutation w/o space

Ex string s = "abc" "abc"

* string s = "abc" "a b c"



"abc"

void Permute(string& a, int l)

```
{ int n = a.size();
  if (l == n)
    { cout << a << endl; return; }
```

```
for (i = l; i < n; i++)
{ swap(a[l], a[i]);
Permute( a, l+1, i );
swap(a[l], a[i]); }
```

}

bac
[] 0

[] ab, 1

[] bac, 1

[] cab, 1

[] abc, 2

[] acb, 2

[] bac, 2

[] acb, 2

[] bac, 3

[] cab, 2

abc

acb

bac, 3

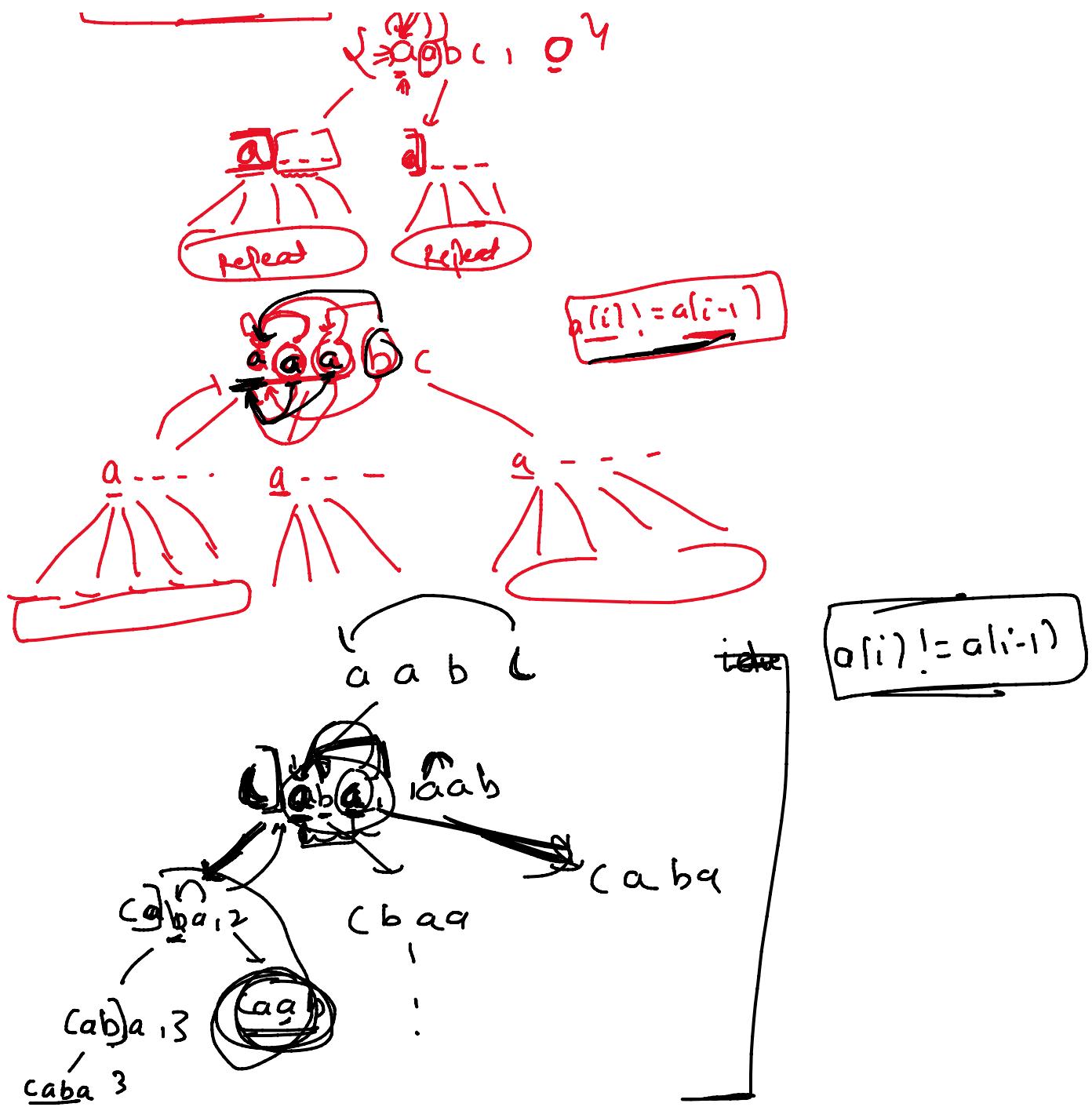
bca, 3

cab, 3

* Duplicates = ?

s = "aabbc"

aabbcc, 0



Subsequence with a sum 1c

[2, 3, 6, 7] → 4 → [2, 3, 6, 7] → 3 → [2, 3, 5, 7] → 7
 [2, 2] → 4 → [2, 3, 6, 7] → 3 → [2, 3, 5, 7] → 7
 $r_1, r_2, r_3, r_4 \Rightarrow [2, 3, 5, 7] + r_4 = 7$

