

Pascal Identity Expansion using Exclude term

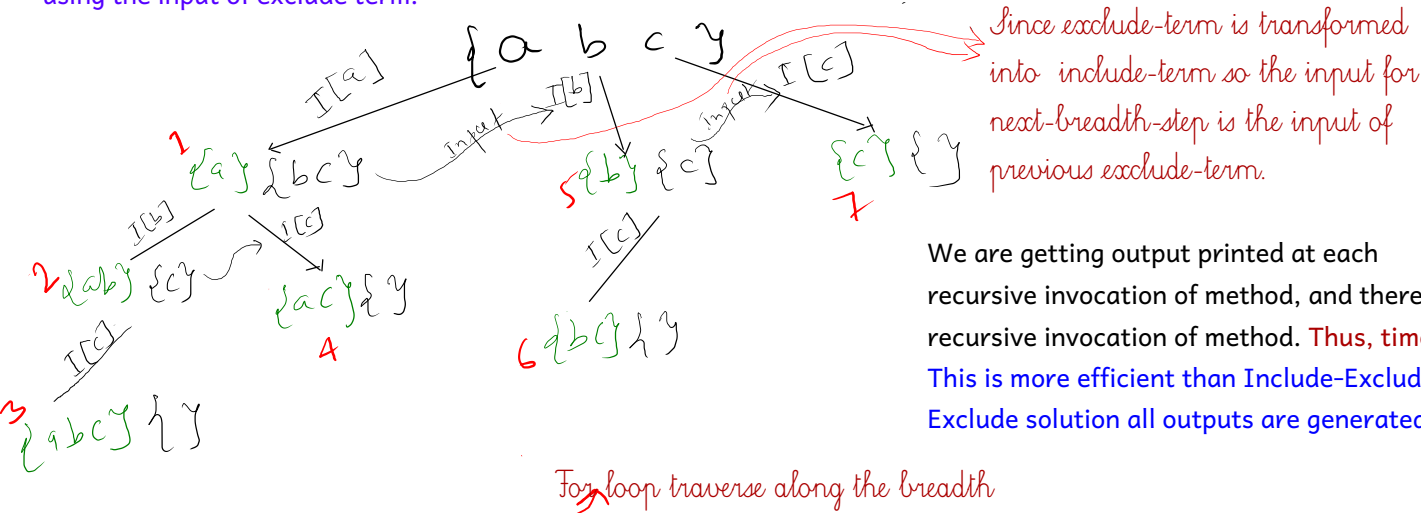
$$n C_k = n-1 C_{k-1} + n-2 C_{k-1} + n-3 C_{k-1} + \dots + k-1 C_{k-1}$$

Expansion is based on recursive expansion of **exclude-term** $C(n-1, k)$ of pascal identity

$$C(n, k) = C(n-1, k-1) + C(n-1, k)$$

$$C(n, k) = C(n-1, k-1) + C(n-2, k-1) + C(n-3, k-1) + \dots C(k-1, k-1)$$

All terms in RHS represents include terms. Exclude term is getting converted in include-term by using the input of exclude term.



We are getting output printed at each recursive invocation of method, and there are 2^n outputs, so 2^n recursive invocation of method. Thus, time-complexity is $O(2^n)$.

This is more efficient than Include-Exclude solution as in Include-Exclude solution all outputs are generated at leaf level.

For loop traverse along the breadth

```
private static void powerSetUsingPascalIdentityExpansion(String input, String output) {
    System.out.println(output);
    for (int i = 0; i < input.length(); i++)
        powerSetUsingPascalIdentityExpansion(input.substring(i + 1), output + input.charAt(i));
}
```