

Print all permutation

Brute force Approach :

Using map. Mark if an index is already taken in ans and do not consider it in recursive call.

- Time Complexity : $O(N! \cdot N)$
- Space Complexity : $O(N)$

Optimised Approach :

Choose an index and perform swap with every other index using a loop. While backtracking make sure to swap again so that the string is unaltered while backtracking.

```
#include <bits/stdc++.h>
void helper(int ind, string& s, vector<string>& ans)
{
    if(ind == s.length())
    {
        ans.push_back(s);
    }
    for(int i = ind; i < s.length(); i++)
    {
        swap(s[ind], s[i]);
        helper(ind+1, s, ans);
        swap(s[ind], s[i]);
    }
}
vector<string> findPermutations(string &s) {
    // Write your code here.
    vector<string> ans;
    helper(0, s, ans);
    return ans;
}
```

- Time Complexity : $O(N! \cdot N)$
- Space Complexity : $O(1)$