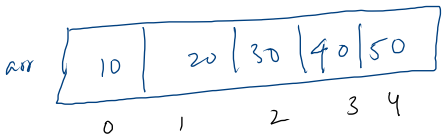
Recursion on the way up | Question / Answer approach

Target Sum Subsets (Day 27)

Problem | Submissions | Leaderboard | Discussions

1. You are given a number n, representing the count of elements.
2. You are given n numbers.
3. You are given a number "tar".
4. Calculate and print all subsets of given elements, the contents of which sum to "tar". Use sample input and output to get more idea.

n = 5



target = 60

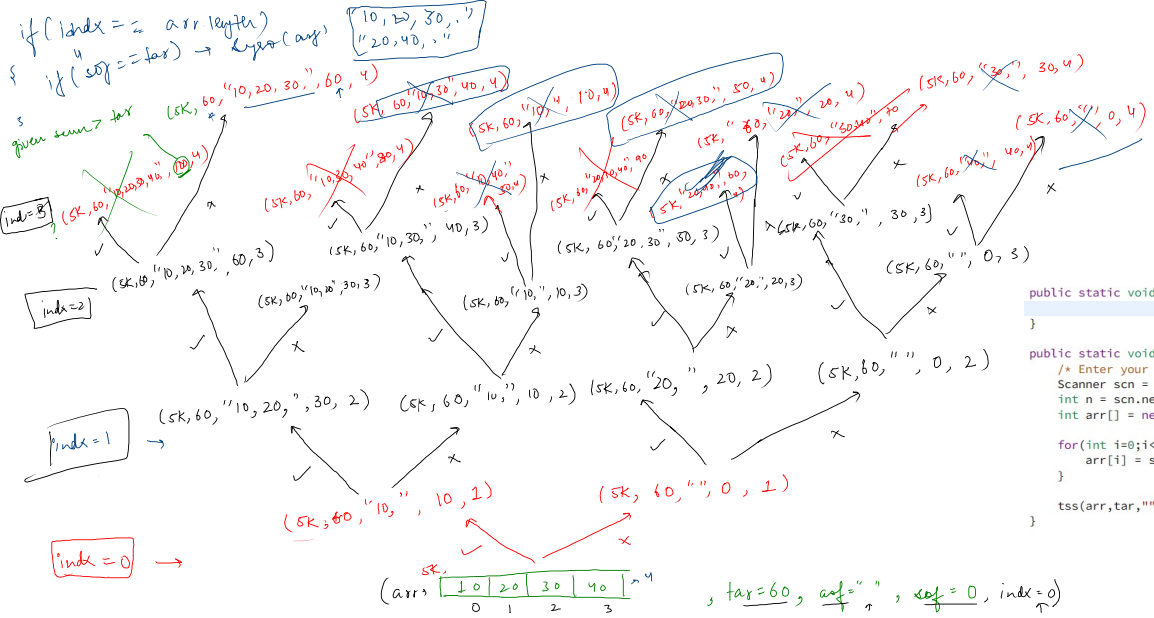
Sample Input 0

```
5
10
20
30
40
50
60
```

Sample Output 0

```
10, 20, 30, .
10, 50, .
20, 40, .
```

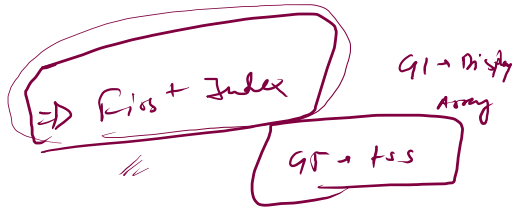
subsequence / subset →




```java
public static void tss(int arr[], int tar, String asf, int sof, int ind){

}

public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT.
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int arr[] = new int[n];

    for(int i=0;i<n;i++){
        arr[i] = scn.nextInt();
    }

    tss(arr,tar,"",0,0);
}
```

Dry Run + Coding

```java
public class Solution {
    public static void tss(int arr[], int tar, String asf, int sof, int ind){
        if(sof> tar){
            return;
        }

        if(ind == arr.length){
            if(sof == tar){
                System.out.println(asf + ".");
            }
            return;
        }

//          Include the current element
        tss(arr,tar, asf+ arr[ind]+", ", sof+arr[ind], ind+1);
//          Exclude the current element
        tss(arr,tar,asf,sof,ind+1);

    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT.
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int arr[] = new int[n];

        for(int i=0;i<n;i++){
            arr[i] = scn.nextInt();
        }

        int tar = scn.nextInt();

        tss(arr,tar,"",0,0);
    }
}
```



Power Linear

$6\ x^n$

$\downarrow$

$x^{n-1} * x$

$\downarrow$

$x^{n-2} * x$

$\downarrow$

$x^0 = 1$

$2^{64} \Rightarrow$ 65 calls

2^{60}

2^{61}

2^{62}

2^{63}

2^{64}

General

$x^n \Rightarrow$ calls $(n+1)$

$64 = 2^{32} * 2^{32} = 2^{64}$

$32 = 2^{16} * 2^{16} = 2^{32}$

$16 = 2^8 * 2^8 = 2^{16}$

$8 = 2^4 * 2^4 = 2^8$

$4 = 2^2 * 2^2 = 2^4$

$2 = 2 \times 2 = 2^2$

$1 * 1 * 2$

$x^n = \begin{cases} x^{n/2} * x^{n/2} & \{ n \text{ is even} \} \\ x^{n/2} * x^{n/2} * x & \{ n \text{ is odd} \} \end{cases}$

65 calls $\Rightarrow 2^{64}$

8 calls $\Rightarrow 2^{64}$

Power log

* Reducing the significant amount of time complexity and space complexity

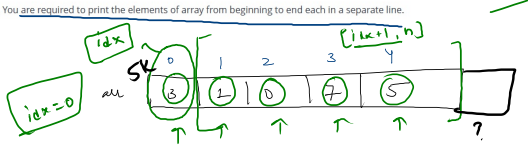
Recursion Basics

→ Recursion with Array

Display Array (Day 24)

Problem | Submissions | Leaderboard | Discussions

1. You are given a number n, representing the size of array a.
2. You are given n numbers, representing elements of array a.
3. You are required to print the elements of array from beginning to end each in a separate line.



n = 5

Sub problem

Sample Input 0

```
5
3
1
0
7
5
```

Sample Output 0

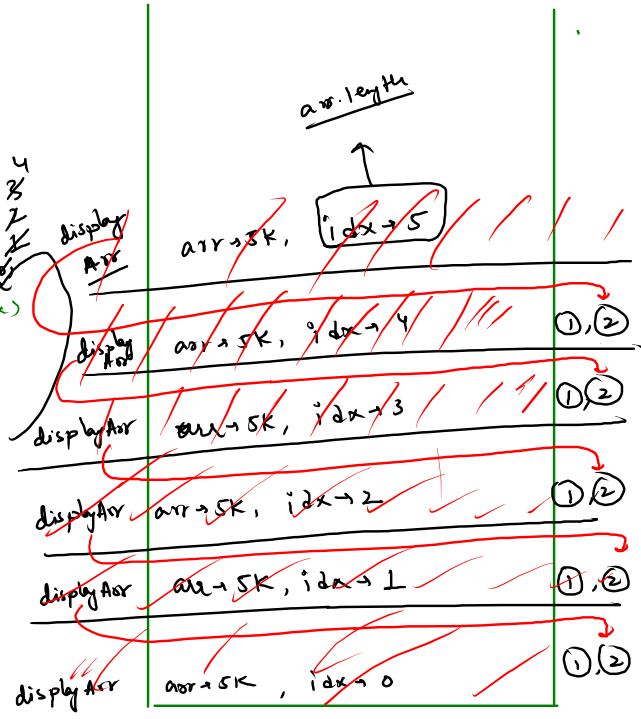
```
3
1
0
7
5
```

```
for(int i=0; i<n; i++)
{
    Sysout(arr[i]);
}
```

```
p.s.  void displayArr(int arr[], int idx)
{
    if(idx == arr.length)
        return;
    Sysout(arr[idx]);          - (1)
    displayArr(arr, idx+1);    - (2)
}
```

3
1
0
7
5

base case : (idx == arr.length)



Recursion with Array

↳ base case → (idx == arr.length || idx == -1)

↳ idx → iterate over all the elements of given array.


```java
import java.io.*;
import java.util.*;

public class Solution {
    public static void displayArr(int arr[], int idx){
        if(idx == arr.length){
            return;
        }


        System.out.println(arr[idx]);
        displayArr(arr, idx+1);
    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print outpu
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int arr[] = new int[n];

        for(int i=0;i<n;i++){
            arr[i] = scn.nextInt();
        }

        displayArr(arr, 0);
    }
}
```

First Index (Day 24)

Problem | Submissions | Leaderboard | Discussions

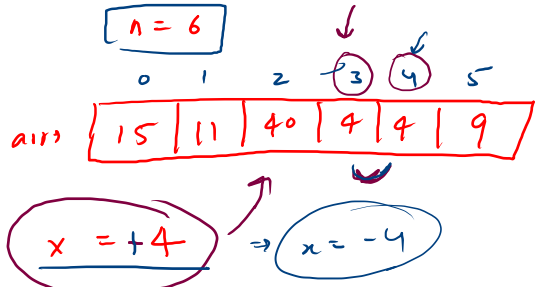
1. You are given a number n, representing the count of elements.
2. You are given n numbers.
3. You are given a number x.
4. You are required to find the first index at which x occurs in array a.
5. If x exists in array, print the first index where it is found otherwise print -1.

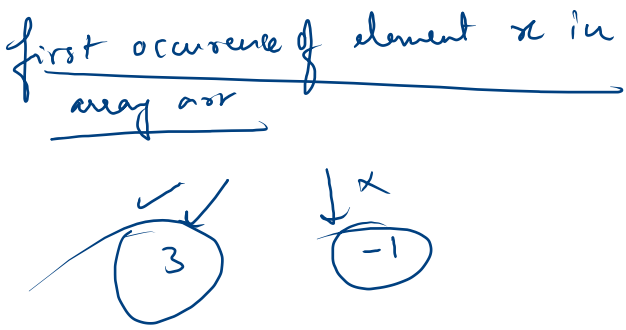
Sample Input 0

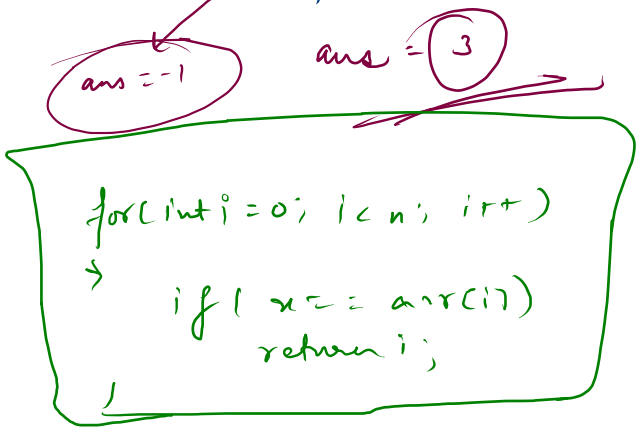
```
6
15
11
40
4
4
9
4
```

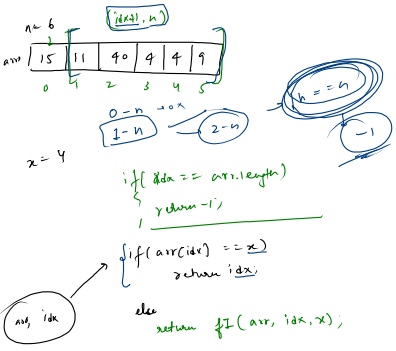
Sample Output 0

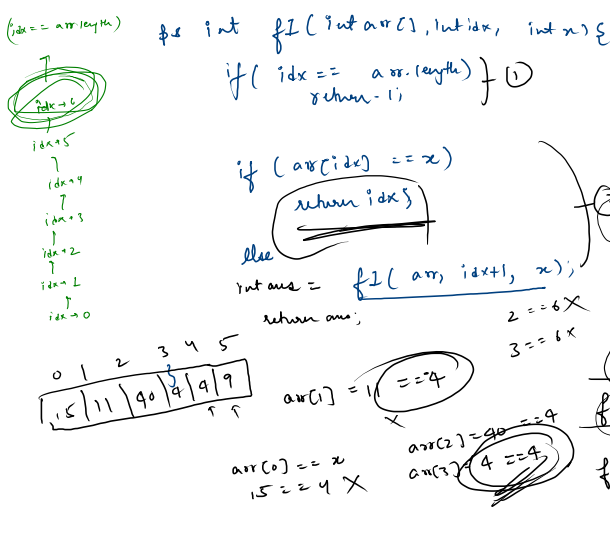
```
3
```

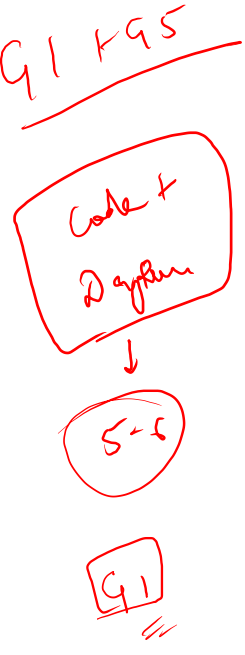




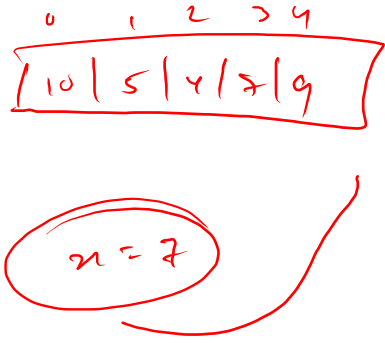


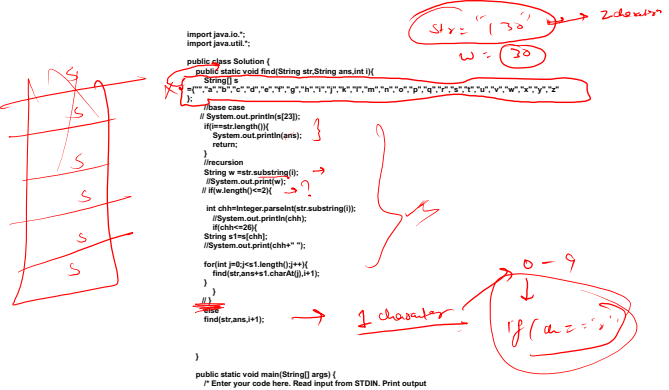

n = 6
(idx+1, n)
arr
15
11
40
4
4
9
0
1
2
3
4
5
0 - n
1 - n
2 - n
x = 4
if(idx == arr.length)
return -1;
if(arr[idx] == x)
return idx;
arr, idx
else
return f1(arr, idx+1, x);


(idx == arr.length)
idx = 6
idx = 5
idx = 4
idx = 3
idx = 2
idx = 1
idx = 0
fs int f1(int arr[], int idx, int x) {
if(idx == arr.length) } ①
return -1;
if (arr[idx] == x)
return idx;
else
int ans = f1(arr, idx+1, x);
return ans;
0 1 2 3 4 5
15 11 40 4 4 9
arr[1] = 11 == 4 X
arr[0] == x
15 == 4 X
arr[2] = 40 == 4
arr[3] = 4 == 4
2 == 6 X
3 == 6 X
f1
arr = 5K, idx = 6, x = 4
ans = -1
arr = 5K, idx = 5, x = 4
ans = -1
arr = 5K, idx = 4, x = 4
ans = -1
arr = 5K, idx = 3, x = 4
ans = 3 -1
arr = 5K, idx = 2, x = 4
ans = 2 -1
arr = 5K, idx = 1, x = 4
ans = 1 -1
arr = 5K, idx = 0, x = 4
① ②
① ②
① ②
① ②
① ②
① ②
3
-1



```java
public class Solution {
    public static int fI(int arr[], int idx, int x){

        if(idx == arr.length){
            return -1;
        }

        if(arr[idx] == x) {
            return idx;
        } else{
            int ans = fI(arr, idx+1, x);
            return ans;
        }
    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int arr[] = new int[n];
        for(int i=0;i<n;i++){
            arr[i] = scn.nextInt();
        }

        int x = scn.nextInt();

        int ans = fI(arr,0,x);
        System.out.println(ans);
    }
}
```



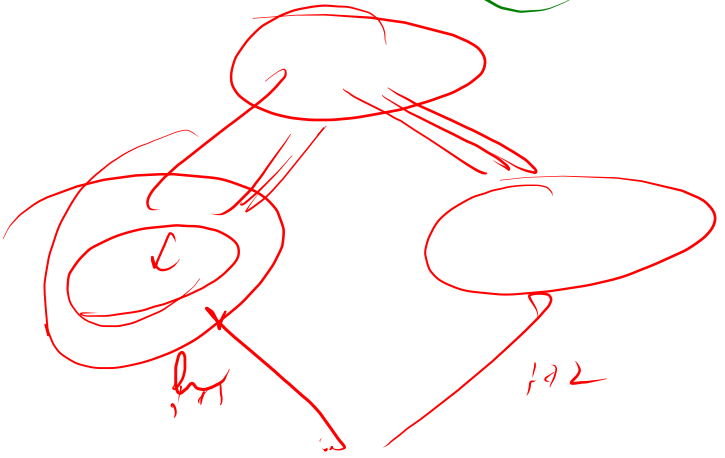


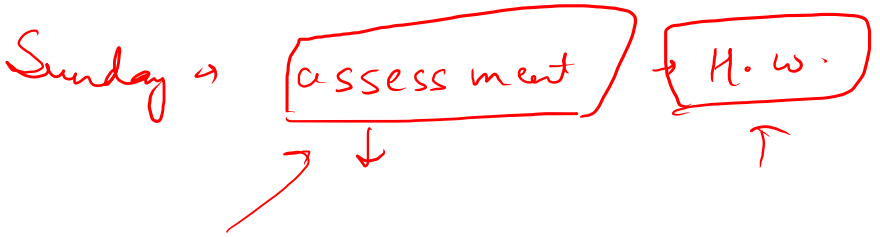
```
import java.io.*;
import java.util.*;

public class Solution {
    public static void find(String str,String ans,int i){
        String[] s
={"","a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r","s","t","u","v","w","x","y","z"
};
        //base case
       // System.out.println(s[23]);
        if(i==str.length()){
            System.out.println(ans);
            return;
        }
        //recursion
        String w =str.substring(i);
         //System.out.print(w);
       // if(w.length()<=2){

          int chh=Integer.parseInt(str.substring(i));
            //System.out.println(chh);
            if(chh<=26){
        String s1=s[chh];
        //System.out.print(chh+" ");

        for(int j=0;j<s1.length();j++){
            find(str,ans+s1.charAt(j),i+1);
        }
          }
       // }
        else
        find(str,ans,i+1);


    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output
```

Iteration → 1-4

Recursion →

Print Decreasing → Dry Run + code




Sunday →
assessment
H.W.


12-1

abc

$3 \times 2 \times 1 = 6$

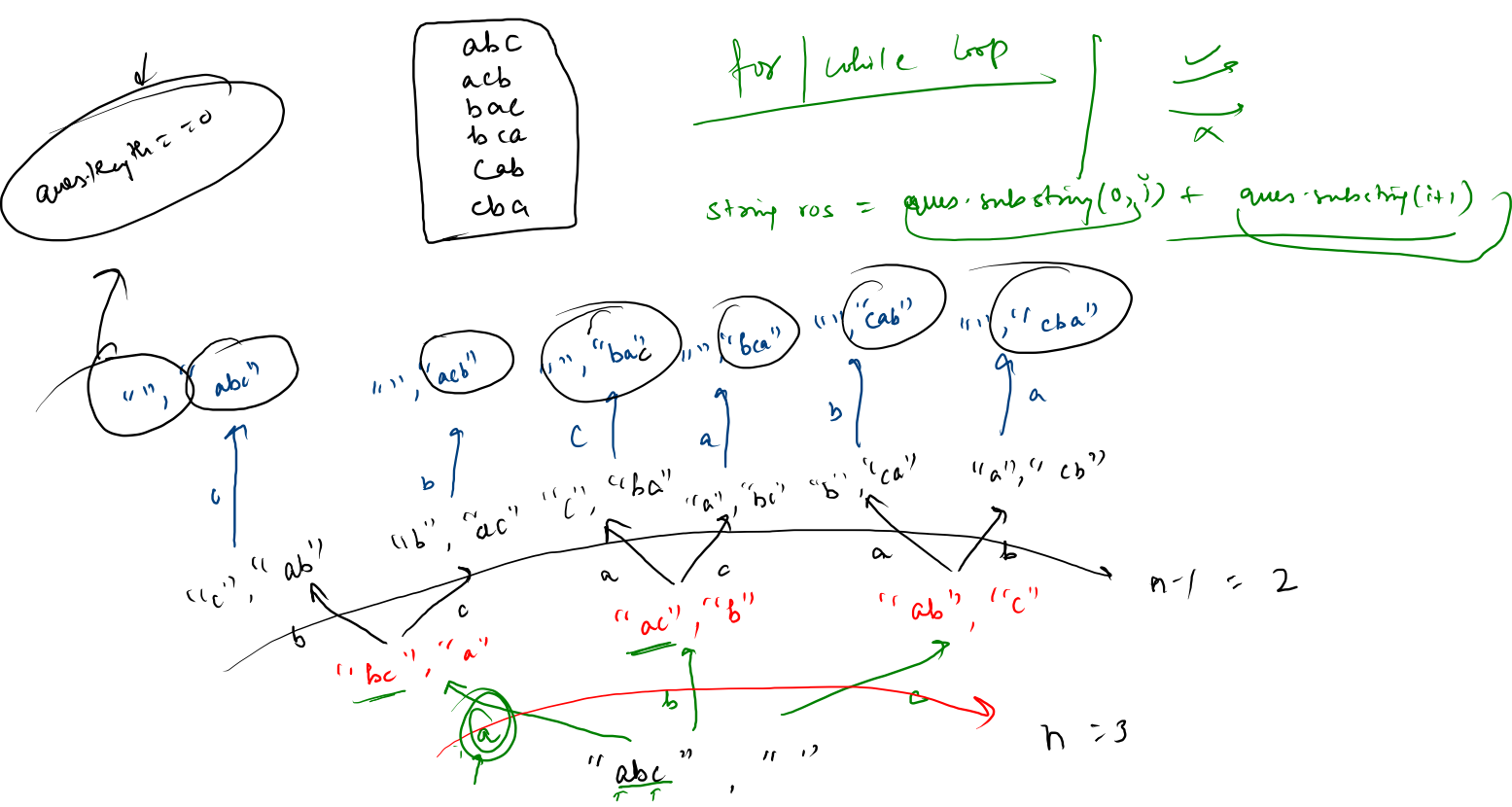
a b c

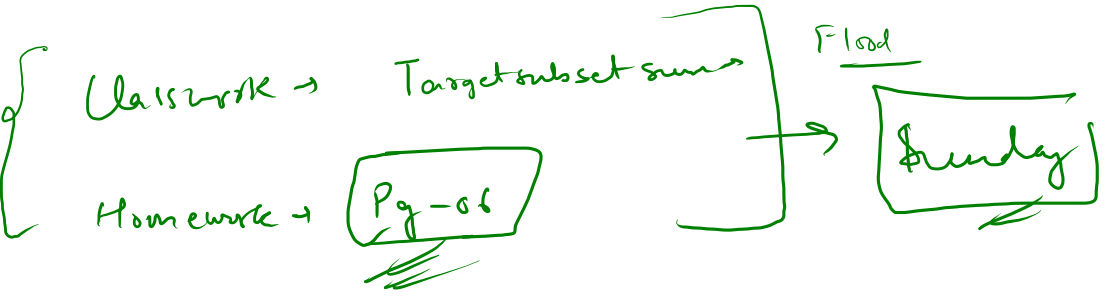
a b c

b c a

n

$n \times (n-1) \times (n-2) \times \cdots$


abc
acb
bac
bca
cab
cba
for | while loop
string ros = ques.substring(0,i) + ques.substring(i+1)
ques.length() == 0
"", "abc"
"", "acb"
"", "bac"
"", "bca"
"", "cab"
"", "cba"
c
b
a
"c", "ab"
"b", "ac"
"c", "ba"
"a", "bc"
"b", "ca"
"a", "cb"
"bc", "a"
"ac", "b"
"ab", "c"
"abc", ""
n-1 = 2
n = 3


Classwork →
Target subset sum
Homework →
Pg - 66
Flood
Sunday