

Power - Linear

$$\underline{x^n} \quad \boxed{x > 0}$$

$$\begin{array}{c} \downarrow \\ x^n \\ \downarrow \\ x^{n-1} \end{array} \quad \begin{array}{c} n=5 \\ \downarrow \\ \text{call } = 6 \\ x^{n-2} \\ \downarrow \\ x^{n-3} \\ | \\ ; \\ x^0 \end{array} \quad \begin{array}{c} 2^5 \\ \diagup \\ T \\ 2^4 \\ \diagup \\ T \\ 2^3 \\ \diagup \\ T \\ 2^2 \\ \diagup \\ T \\ 2^1 \\ \diagup \\ r \\ 2^0 \\ \diagup \\ T \end{array} \quad = \underline{n+1}$$

2⁶⁴ → 65

Power-logarithmic (Day 23)

Problem

Submissions

Leaderboard

Discussions

1. You are given a number x .
2. You are given another number n .
3. You are required to calculate x raised to the power n .



A handwritten mathematical expression where the letter 'x' is followed by a superscript 'n'. The entire expression is enclosed within a hand-drawn circle.

$$x^n$$

$$\begin{array}{c} \textcircled{36} \\ \textcircled{2} \\ \downarrow \\ \textcircled{18} \in a \\ \downarrow \\ \textcircled{9} \end{array} \Rightarrow 2^{36} = 2^{18} \times 2^{18}$$

$$\frac{36}{2} = 18$$

$$x=2$$

$$2^{34}$$

if (power - even)
 b ~~*~~ a * a = ans
 else
 b a * a * x

$$\begin{array}{c} \textcircled{9} \\ \textcircled{2} \\ \downarrow \\ \textcircled{9} \end{array} \quad 2^{18} = 2^9 \times 2^9 = 2^{18}$$

$$\frac{9}{2} = \textcircled{4}$$

$$\begin{array}{c} \textcircled{4} \\ \textcircled{2} \\ \downarrow \\ \textcircled{4} \end{array} \Rightarrow 2^4 \times 2^4 = 2^8 \times 2^4 = 2^8$$

$$\begin{aligned}
 & 2^{69} \leftarrow \\
 & 2^{34} \rightarrow 2^{17} \times 2^{17} = \boxed{2^{34}} \\
 & \downarrow \\
 & 2^{17} \rightarrow 2^8 \times 2^8 \times 2^{17} = 2^{34} \\
 & \downarrow \\
 & 2^8 \rightarrow 2^4 \times 2^4 = 2^8 \\
 & \downarrow \\
 & 2^4 \rightarrow 2^2 \times 2^2 = 2^4 \\
 & \downarrow \\
 & 2^2 \rightarrow 2 \times 2 = 2^2 \\
 & \downarrow \\
 & \textcircled{1} \rightarrow 1 \times 1 + 1 \times 2 = 2 \\
 & \downarrow \\
 & 2^0 \rightarrow 1
 \end{aligned}$$

base case

$$\begin{aligned}
 & 2^{34} \rightarrow \text{power linear} \rightarrow \text{cells} = 34 + 1 = \boxed{35} \\
 & \text{power log.} \rightarrow \text{cells} = \cancel{?} \\
 & \Rightarrow \textcircled{20} \text{ pass cells}
 \end{aligned}$$

$$\begin{cases}
 1) \text{Expectation} \rightarrow x^n \rightarrow 2^{34} \\
 2) \text{Fai th} \rightarrow x^{\frac{n}{2}} \\
 3) \text{Buy work} \rightarrow
 \end{cases}
 \begin{array}{l}
 \xrightarrow{n \text{ is even}} x^{\frac{n}{2}} * x^{\frac{n}{2}} \\
 \xrightarrow{n \text{ is odd}} x^{\frac{n-1}{2}} * x^{\frac{n+1}{2}} * x
 \end{array}$$

4) base case \rightarrow if ($n = 0$)

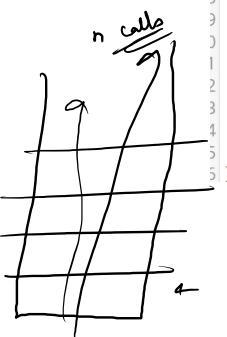
return 1

Expectation \rightarrow Question

```

7 public class Solution {
8     public static int powerLog(int x, int n){
9         if(n == 0) return 1;
10
11         int subAns = powerLog(x,n/2);
12         int ans = subAns*subAns;
13         if(n%2 != 0) ans*=x;
14
15         return ans;
16     }
17
18     public static void main(String[] args) {
19         /* Enter your code here. Read input from STDIN. Print
20            Scanner scn = new Scanner(System.in);
21            int x = scn.nextInt();
22            int n = scn.nextInt();
23
24            System.out.println(powerLog(x,n));
25        }

```



$$\sqrt{5 + 5} \quad \text{min } \rightarrow \text{array}$$

Dodgy
me

power log $\rightarrow T.C \rightarrow O(\log^6)$
 $S.C \rightarrow O(\log^6)$

power linear $\rightarrow T.C = O(n)$
 $S.C \rightarrow O(n)$ (due to recursion stack)



{ \Rightarrow constant }

$$T(n) = T(n/2) + k \quad \rightarrow \text{Recurrence relation}$$

$\log_2 n$

$\overbrace{(n+n)}^{\text{n times}} \approx \cdot$

Display Array (Day 24)

$n=5$

Problem

Submissions

Leaderboard

Discussions

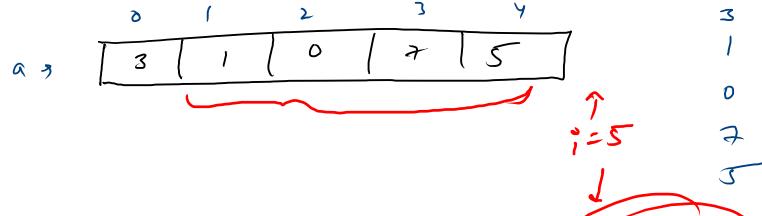
1. You are given a number n , representing the size of array a .

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.

Recursive

$n=5$



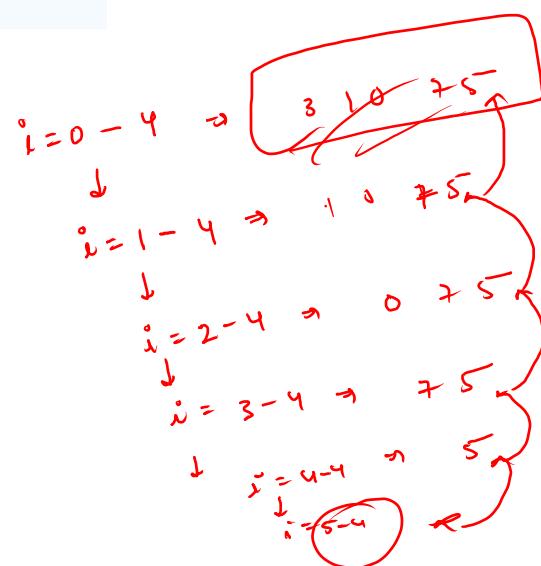
- 1) expectation: $(i, n-1) \Rightarrow 3 | 1 | 0 | 7 | 5$
- 2) faith: $(i+1, n-1) \Rightarrow 1 | 0 | 7 | 5$
- 3) My work \rightarrow print the 1st index element $0 | 7 | 5$
- 4) base case \hookrightarrow if ($i == n$) return

Sample Input 0

" → 5
3
1
0
7
5

Sample Output 0

3
1
0
7
5



$\rightarrow S + SC$

```

public class Solution {
    public static void displayArray(int index, int n, int arr[]){
        if(index == n) return; -①
        System.out.println(arr[index]); -②
        displayArray(index+1,n,arr); -③
    }

    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();
        }

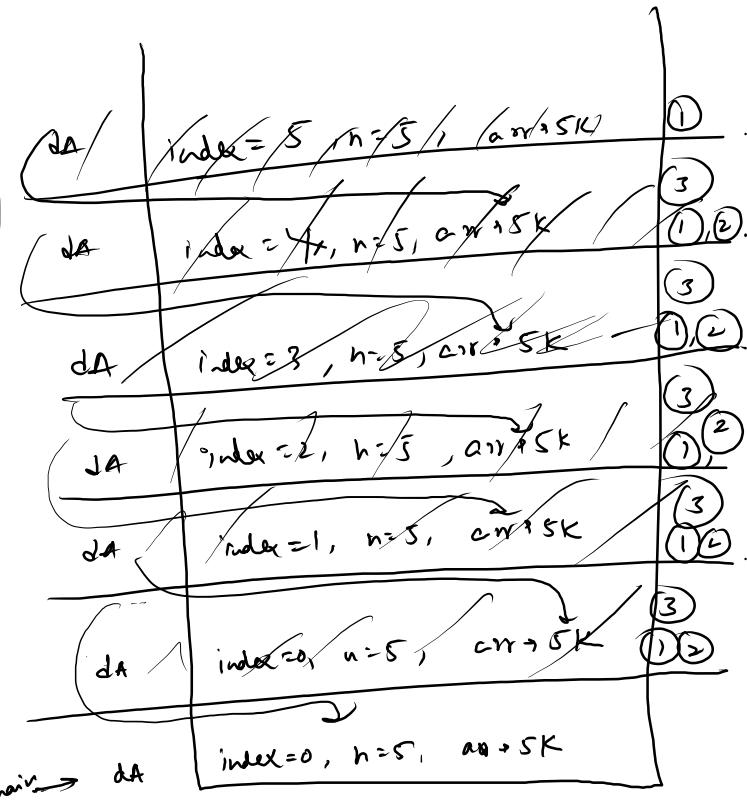
        displayArray(0, n, arr);
    }
}

```



Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Ca

$n=5$



First Index (Day 24)

Problem

Submissions

Leaderboard

Discussions

Sample Input 0

6
15
11
40
4
4
9
→ 4

1. You are given a number n, representing the count of elements.

2. You are given n numbers.

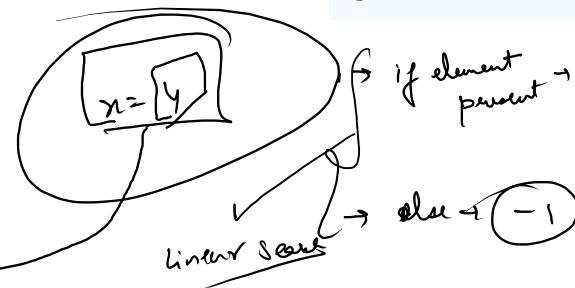
3. You are given a number x.

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.

Note -> The online judge can't force you to write the function recursively but that is what the spirit of question is. Write recursive and not iterative logic. The purpose of the question is to aid learning recursion and not test you.

$$n = 6$$



Sample Output 0

3

- 1) Expectation → Find the first occurrence of $x \rightarrow$ from 0 to n
- 2) Faith → Find the first occurrence of $x \rightarrow$ from 1 to n
- 3) My work → if ($\text{arr}[i] == x$) return i ;
- 4) base case → if ($i == n$) return -1

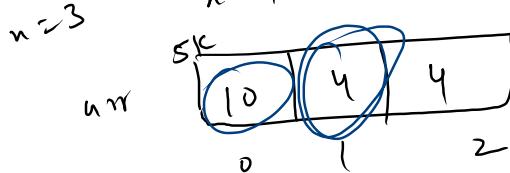

```

public static int firstIndex(int i, int n, int arr[], int x) {
    if(i == n) return -1;
    if(arr[i] == x) return i;
    else {
        int ans = firstIndex(i+1, n, arr, x);
        return ans;
    }
}

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 x = scn.nextInt();
    int ans = firstIndex(0, n, arr, x);
    System.out.println(firstIndex(0, n, arr, x));
}

```

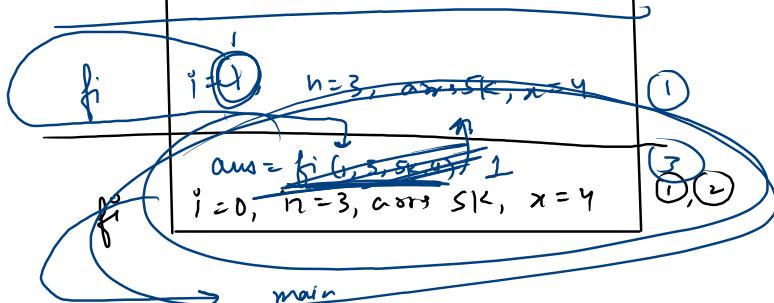


$$0 == 3$$

$$1 == 3 \times$$

$$4 == 4$$

$$10 == 4$$



All Indices Of Array (Day 24)

Problem

Submissions

Leaderboard

Discussions

1. You are given a number n, representing the count of elements.

2. You are given n numbers.

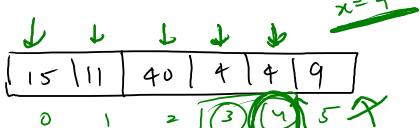
3. You are given a number x.

4. You are required to find all indices at which x occurs in array a.

5. Return an array of appropriate size which contains all indices at which x occurs in array a.

int

$n = 6$



Sample Input 0

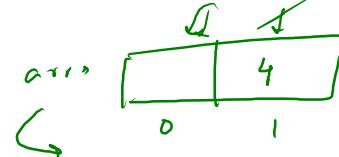
$n \rightarrow$
6
15
11
40
4
4
9
4

Sample Output 0

3
4

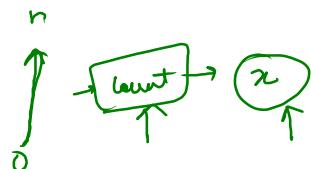
at base case →

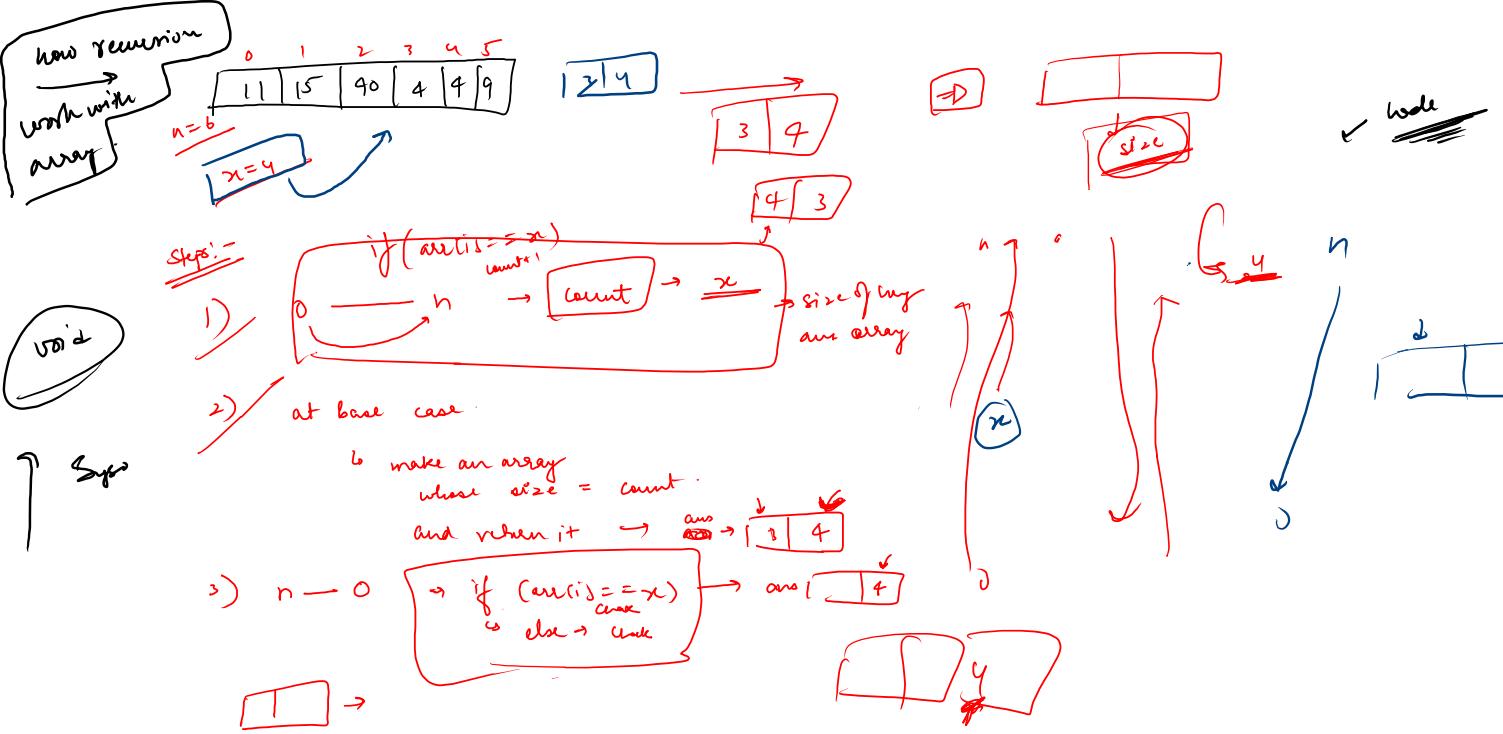
make an array of size
= count - 2



int arr[] → return

for (int val; arr)
 sys(val);





```

public class Solution {
    public static int[] findAllIndices(int ind, int n, int arr[], int x, int count) {
        if(ind == n){
            int base[] = new int[count];
            return base;
        }

        int ans [] ;
        if(arr[ind] == x){
            ans = findAllIndices(ind+1,n,arr,x,count+1);
            ans[count] = ind;
        }else{
            ans = findAllIndices(ind+1,n,arr,x,count);
        }

        return ans;
    }
}

```

$$n = 6$$

$$15 = 4x$$

$$11 = 4x$$

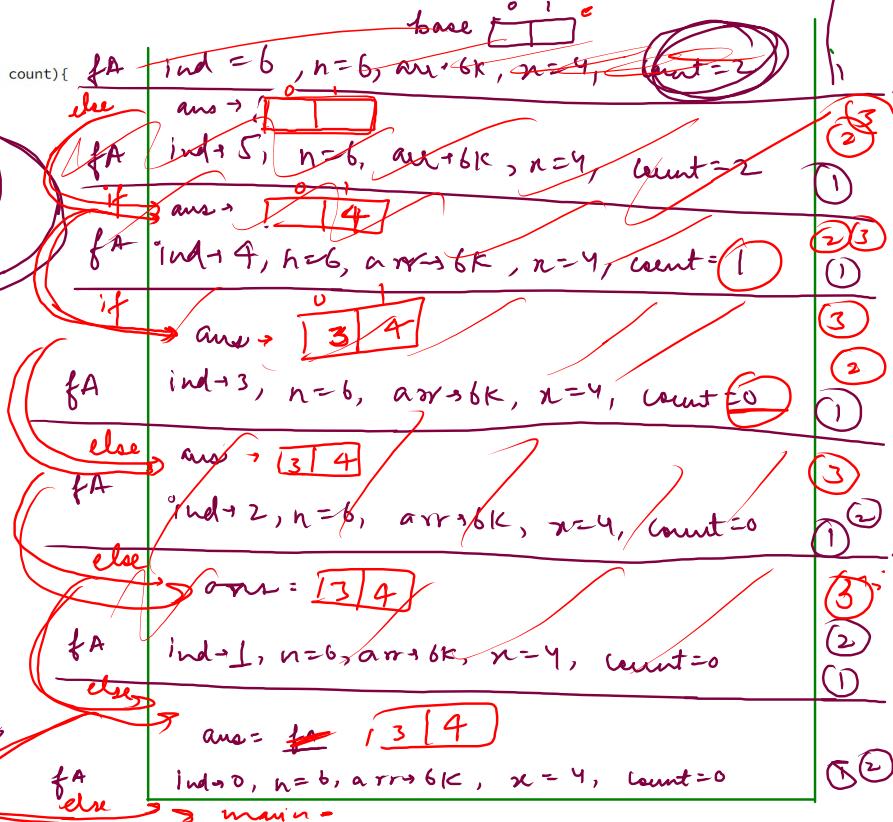
0	1	2	3	4	5
15	11	40	4	4	9

$$x = 4$$

$$9 = 4x$$

$$90 = 4x$$

$$4 = 4x$$



```
public class Solution {  
    public static int [] findAllIndices(int ind,int n, int arr[], int x, int count){  
        if(ind == n){  
            int base[] = new int[count];  
            return base;  
        }  
  
        int ans [] ;  
        if(arr[ind] == x){  
            ans = findAllIndices(ind+1,n,arr,x,count+1);  
            ans[count] = ind;  
        }else{  
            ans = findAllIndices(ind+1,n,arr,x,count);  
        }  
  
        return ans;  
    }  
  
    public static void main(String[] args) {  
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be  
        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 [] = findAllIndices(0,n,arr,x,0);  
        for(int i=0;i<ans.length;i++){  
            System.out.println(ans[i]);  
        }  
    }  
}
```

String

Subsequence / Subset :-

↳ part of string

str = "abcde"

~~ab cde~~

→
a
ab
abc
abcd
abcde

str = "abc"



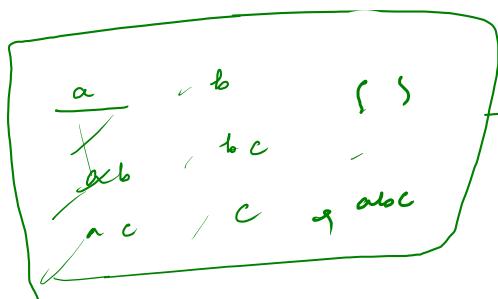
Length of the string = n

Total subsequence = $2 \times 2 \times 2 \times \dots$ n times

$$= \boxed{2^n}$$

⇒ substring / subarray is a subset of subsequence

$$abc \rightarrow 2^3 = 8$$



subsequence of given string "abc"

a
ab
abc

b
bc

c

Get Subsequence (Day 24)

CONTENTS OF THE ARRAYLIST CONTAINING SUBSEQUENCES

Problem

Submissions

Leaderboard

Discussions

1. You are given a string str.

2. Calculate all subsequences of str. Use sample input and output to take idea about subsequences.

Sample Input 0

abc

Sample Output 0

`[[], c, b, bc, a, ac, ab, abc]`

$$\text{str} = "abc" \rightarrow 2^3 = 8$$

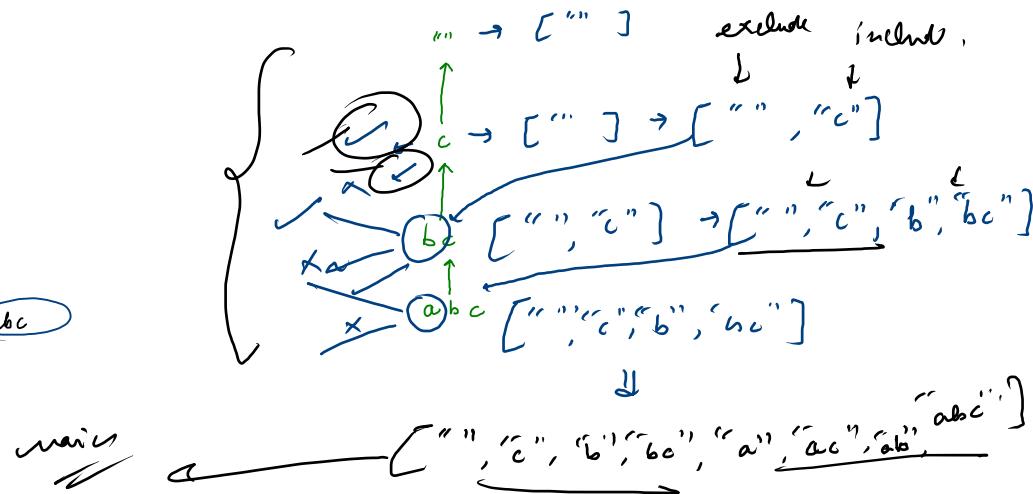
(exclude, include) ArrayList

- 1) expectation → Generate all the subsequence of "abc"
- 2) faith → Generate all the subsequence of "bc"
- 3) My work → exclude a, exclude a
- 4) base case:- `str.length() == 0` → Arraylist

`(i-1)`

✓ + - -
x

abc



```

public class Solution {
    public static ArrayList<String> getSub(String str){
        if(str.length() == 0){
            ArrayList<String> base = new ArrayList<>();
            base.add("");
            return base;
        }

        char ch = str.charAt(0);
        String ros = str.substring(1);
        ArrayList<String> subAns = getSub(ros); } → (1)

        ArrayList<String> ans = new ArrayList<>();
        // Exclude
        for(int i=0;i<subAns.size();i++){
            ans.add(subAns.get(i));
        }

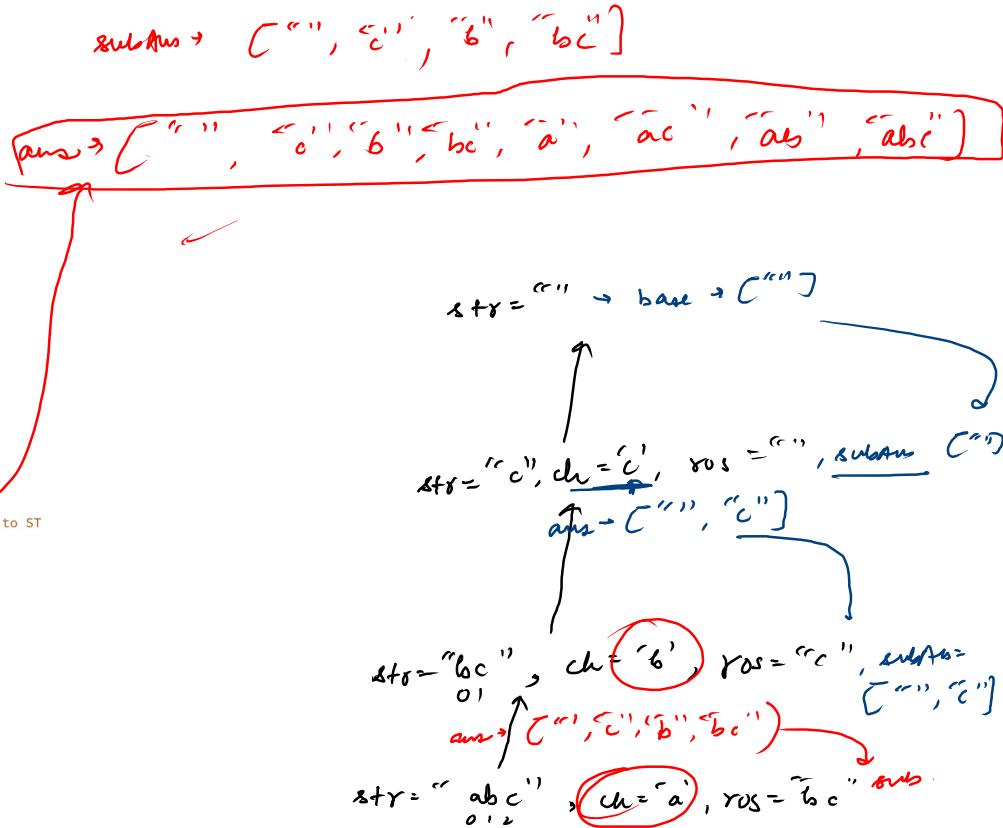
        // Include
        for(int i=0;i<subAns.size();i++){
            ans.add(ch+subAns.get(i));
        }

        return ans;
    }

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

        System.out.println(getSub(str));
    }
}

```



```

public class Solution {
    public static ArrayList<String> getSub(String str){
        if(str.length() == 0){
            ArrayList<String> base = new ArrayList<>();
            base.add("");
            return base;
        }

        char ch = str.charAt(0);
        String ros = str.substring(1);
        ArrayList<String> subAns = getSub(ros);
    }

    ArrayList<String> ans = new ArrayList<>();
    // Exclude
    for(int i=0;i<subAns.size();i++){
        ans.add(subAns.get(i));
    }

    // Include
    for(int i=0;i<subAns.size();i++){
        ans.add(ch+subAns.get(i));
    }

    return ans;
}

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

    System.out.println(getSub(str));
}

```

ans + ["", "c", "b", "bc", "a", "ac", "ab", "abc"]

1, 2, ③

1, 2,

1, 2, ③

1, 0, num → code + dry run

~~str = "c", base = [""]~~

~~str = "c", base = [""]~~

~~str = "c", base = [""]~~, subAns = [""]

~~ans = ["", "c"]~~

~~str = "bc", ch = "b", ros = "c", subAns = ["", "c"]~~

~~ans = ["", "c", "b", "bc"]~~

~~str = "abc", ch = "a", ros = "bc", subAns = ["", "c", "b", "bc"]~~

↓ main

Get Stair Paths (Day 25)

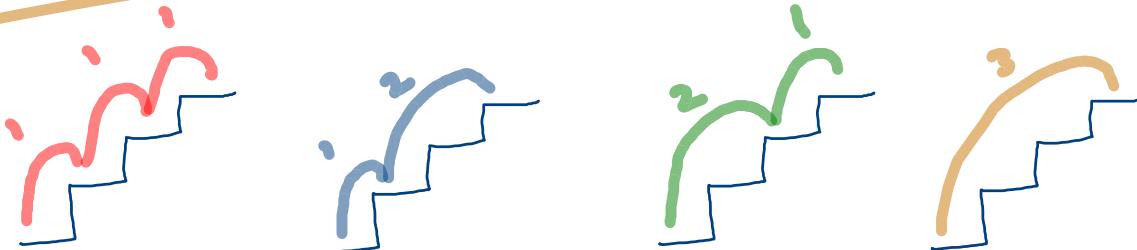
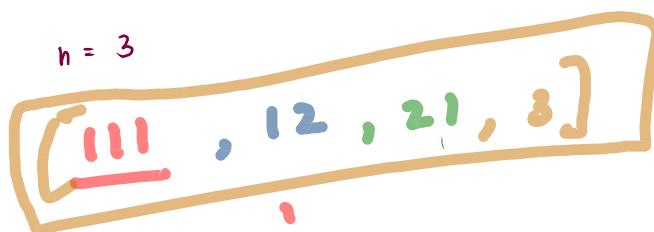
Sample Input 0

3

1. You are given a number n representing number of stairs in a staircase.
2. You are standing at the bottom of staircase. You are allowed to climb 1 step, 2 steps or 3 steps in one move.
3. Get the list of all paths that can be used to climb the staircase up.

Sample Output 0

[111, 12, 21, 3]



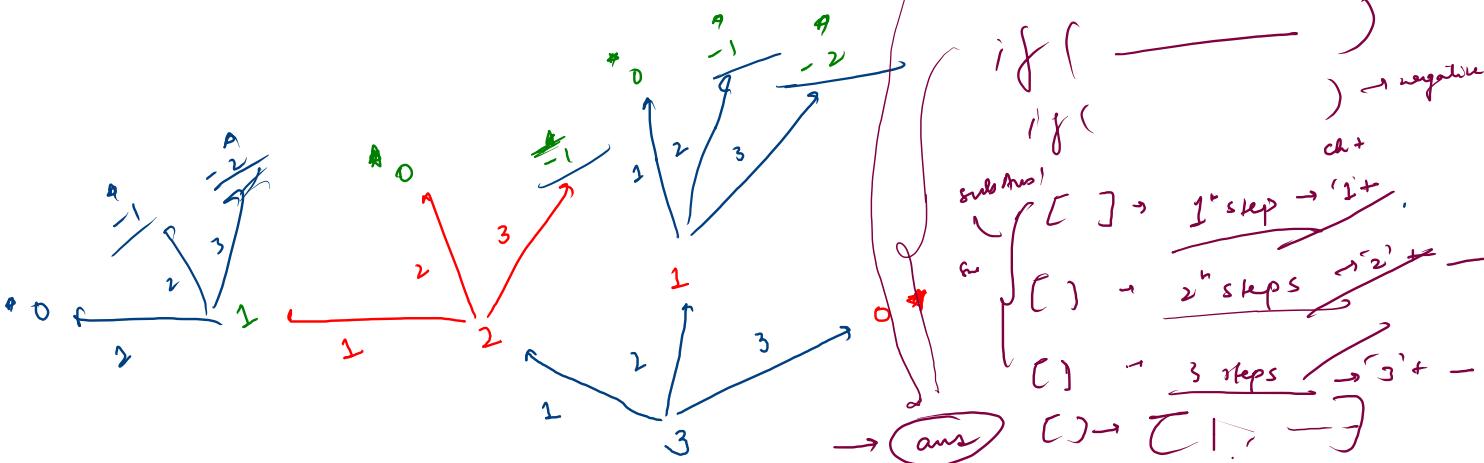
1 recursive \rightarrow 1 step

1 recursive \rightarrow 2 steps

1 recursive \rightarrow 3 steps

3 recursive calls, 2 base cases.

base case 1 \rightarrow destination \rightarrow positive base case
base case 2 \rightarrow wrong destination \rightarrow negative base case



```

public static ArrayList<String> getStairPaths(int n){
    if(n<0){ // negative base case
        return new ArrayList<>();
    }
    if(n == 0){ // positive base case
        ArrayList<String> base1 = new ArrayList<>();
        base1.add("");
        return base1;
    }
}

```

```

ArrayList<String> subAns1 = getStairPaths(n-1); // 1 step → (1)
ArrayList<String> subAns2 = getStairPaths(n-2); // 2 steps → (2)
ArrayList<String> subAns3 = getStairPaths(n-3); // 3 steps → (3)

ArrayList<String> ans = new ArrayList<>();
for(int i=0;i<subAns1.size();i++){
    ans.add("1"+subAns1.get(i));
}
 $\dots$ 
for(int i=0;i<subAns2.size();i++){
    ans.add("2"+subAns2.get(i));
}

for(int i=0;i<subAns3.size();i++){
    ans.add("3"+subAns3.get(i));
}

return ans;
}

```

```

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

    System.out.println(getStairPaths(n));
}

```

$ans \rightarrow \underline{\underline{["11", "2"]}}$

Coding, Dry run
NW
Dry run

'5'

'5-6'
'6'

'5'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

'6'

Print Alternate Column wise(14 July)

Problem

Submissions

Leaderboard

Discussions

Print the matrix column wise such that we print the alternate columns of the matrix starting from the first column.

$$n=3, m=3$$

0	0	1	2
0	49	22	16
1	71	20	23
2	89	24	61

0 1 2 - 2

49	71	89
16	23	61

Sample Input 0

```
3
3
49 22 16
71 78 23
89 24 61
```

Sample Output 0

```
49 71 89
16 23 61
```

```
for (int j = 0; j < m; j += 2) {
    for (int i = 0; i < n; i++) {
        cout << arr[i][j] << " "
    }
}
```

0 0
0 +
1 0
20

Rotate An Array (16 july)

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 given a number k .

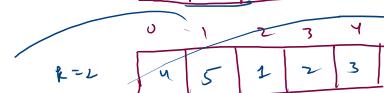
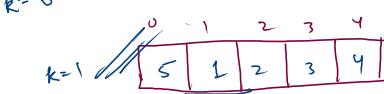
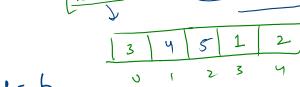
4. Rotate the array a , k times to the right (for positive values of k), and to the left for negative values of k .

Sample Input 0

```
5  
1  
2  
3  
4  
5
```

Sample Output 0

```
3 4 5 1 2
```



$k > n$

$k=4$

$k \equiv 0 \pmod{n}$

$\text{if } (k > n) \quad \text{or} \quad k = k \pmod{n};$

$\text{if } (k < 0) \quad k = k + n \Rightarrow$

$\text{if } (k < 0) \quad k + n \equiv 1$

$\text{if } (k < 0) \quad k + n \equiv 2$

$k = k \pmod{n}$

$k = -2 + 5 = 3$

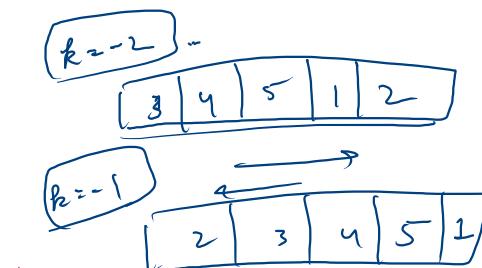
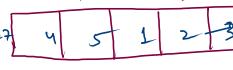
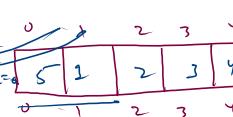
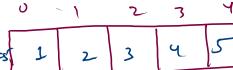
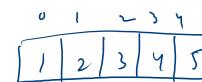
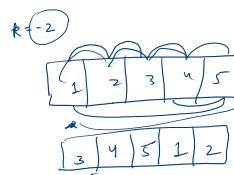
$k = -1 + 5 = 4$

$k = -3 + 5 = 2$

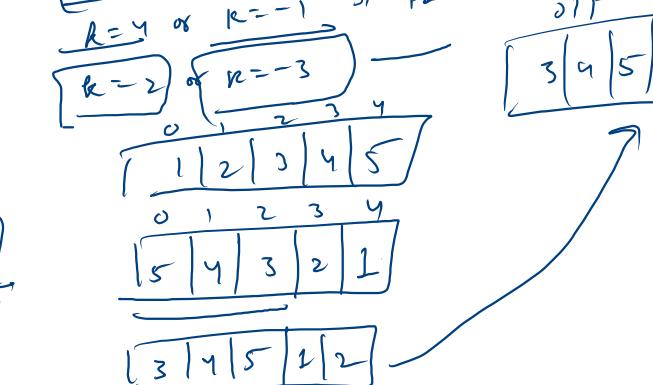
reverse the array $(0, n-1)$

reverse the array $(0, k-1)$

reverse the array $(k, n-1)$



$k=3$ or $k=-2 \Rightarrow$ Pattern is same.



```

public class Solution {
    public static void reverse(int arr[], int left, int right){
        while(left<right){
            int temp = arr[left];
            arr[left] = arr[right];
            arr[right] = temp;

            left++;
            right--;
        }
    }

    public static void rotateArray(int arr[],int n,int k){
        if(k<0) k+=n;
        if(k>n) k%=n;

        reverse(arr,0,n-1);
        reverse(arr,0,k-1);
        reverse(arr,k,n-1);
    }

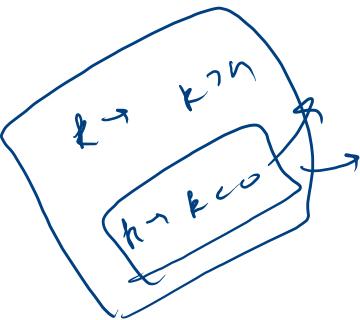
    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT */
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();

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

        rotateArray(arr,n,k);

        for(int i=0;i<n;i++){
            System.out.print(arr[i] + " ");
        }
    }
}

```



Target String (22 july)

Problem

Submissions

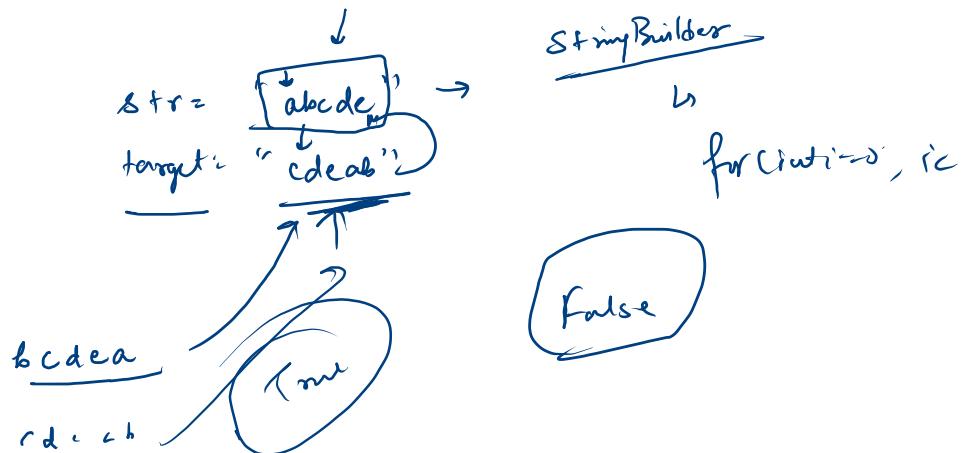
Leaderboard

Discussions

Sample Input 0

abcde
cdeab

Take Two Strings as input. First string as "str" and second string as a "Target" string. You are allowed to rotate the original string "str" multiple times. Print "True" if "Target" string can be achieved by rotating the "str" any number of times, else print "False". Note: String "bcda" is a rotation of "abcd" but "bdca" is not a rotation of String "abcd"



Sample Output 0

True

{
 reverse(0, n-1)
 reverse(0, i-1)
 reverse(i, n-1)}

Find Distance B/W Two Characters (23 july)

Sample Input 0

Geeks
G
S

Problem

Submissions

Leaderboard

Discussions

last character occurrence

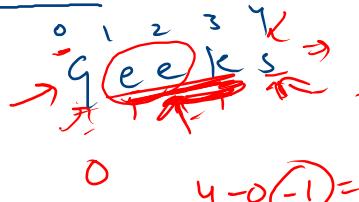
Given a string and two characters. Print the minimum distance between two given characters a and b in a string.

$$s = "Geeks"$$

ch₁ = 'G'
ch₂ = 'S'

$$ch_1 = 'e' \rightarrow \text{Min} \rightarrow$$

$$ch_2 = 'k'$$



$$G \rightarrow 0$$
$$S \rightarrow 4$$

$$4 - 0 - 1 = 3$$

$$\begin{array}{ccccccc} & & & & 3-1-1-1 \\ l & \rightarrow & k & \rightarrow & 3-1-1-1 \\ e & \rightarrow & k & \rightarrow & 3-2-1-0 \\ r & \rightarrow & k & \rightarrow & 3-2-1-0 \\ e & \rightarrow & k & \rightarrow & 3-2-1-0 \\ \hline & 2 & 3 & 3-2+1=0 & \end{array}$$

Sample Output 0

3

g cokse
es

Geeks

ch1 = C
ch2 = R

→ String

for

lastOccurrence

ans

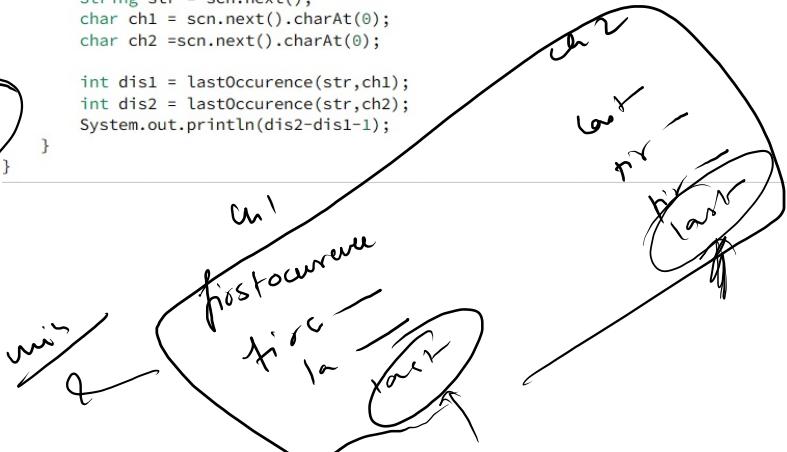
dis1 = 2
dis2 = 3
dis = 3 - 1 - 1 = 1

dis = 3 - 2 - 2 = 0

```
public class Solution {
    public static int lastOccurrence(String str, char ch){
        int ans = -1;
        for(int i=0; i<str.length(); i++){
            if(ch == str.charAt(i)) ans = i;
        }
        return ans;
    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be in
        Scanner scn = new Scanner(System.in);
        String str = scn.next();
        char ch1 = scn.next().charAt(0);
        char ch2 = scn.next().charAt(0);

        int dis1 = lastOccurrence(str, ch1);
        int dis2 = lastOccurrence(str, ch2);
        System.out.println(dis2-dis1-1);
    }
}
```



920. Number of Music Playlists



Hard 2.1K 179 ⭐ ⓘ

Companies

Your music player contains `n` different songs. You want to listen to `goal` songs (not necessarily different) during your trip. To avoid boredom, you will create a playlist so that:

- Every song is played **at least once**.
- A song can only be played again only if `k` other songs have been played.

Given `n`, `goal`, and `k`, return *the number of possible playlists that you can create*. Since the answer can be very large, return it **modulo** $10^9 + 7$.

Example 1:

Input: `n = 3, goal = 3, k = 1`

Output: 6

Explanation: There are 6 possible playlists: [1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], and [3, 2, 1].

Example 2:

Input: `n = 2, goal = 3, k = 0`

Output: 6

Explanation: There are 6 possible playlists: [1, 1, 2], [1, 2, 1], [2, 1, 1], [2, 2, 1], [2, 1, 2], and [1, 2, 2].

Example 3:

Input: `n = 2, goal = 3, k = 1`

Output: 2

Explanation: There are 2 possible playlists: [1, 2, 1] and [2, 1, 2].

$$n = 3 \quad \text{goal} = 3$$