1. ­-Reverse a Linked List:
2. If a person can climb to stairs in one go as 1 stair or 2 stairs , find the ways person can climb to the nth stair?

Use below formula to calculate the ways(n)

Ways(n)=way(n-1) + ways (n-2)

The above formula is for the formula for the febonicii number.

But the value of ways(n) is equal to the fibonicii(n+1).

1. A number and string can be palindrome if there reverse is also same.
2. For number palindrome check take the modulus of the number by 10 and store as remainder. Do it through the loop till num>0.
3. For string palindrome check, take char at str.length()-1 and do the substring always from 0 to length-1. i.e str.subString(0,str.length()-1). Which exclude upper limit..
4. String permutation programme hint
5. Define a method with argument Str, left index (0), right index (str.length()-1).
6. Check if left and right index equal return the string.
7. Else loop through left index to till right index
8. Define a method swap with argument str, left and i index of loop and call this from loop.
9. Swap method first convert the string to char array and then value at left index store in temp variable and value at i index store in left index. And temp value store at I index. Convert char array to string and return. This method will return the string which will pass to the permutateString method.
10. Call recursively permutateString with argument str,left+1 and right index. Indside the loop.
11. Call again swap method inside loop with same argument as called in point #iv.
12. Array missing number hint
13. This we can find out using the binary search as it has a O (log n ) time complexity.

Or

1. First define total array length by adding missing numbers count in the current array length.
2. Find the sum of the two missing number by using below formula

int sum=n\*(n+1)/2 – sum of the all array current elements

1. Find the average of the two missing numbers

int avg=sum/2;

1. Find the sum of the array element which are smaller or equal to avg and sum of the array element which are greater than avg.
2. Find the first missing element by using the below formula

// The first (smaller) element =

    // (sum of natural numbers upto

    // avg) - (sum of array elements

    // smaller than or equal to avg)

Int firstElement=avg\*(avg+1)/2 + sum of element from #iv

Second missing element

= (n\*(n+1)/)-total smaller half – sum of greater half from #v

1. Check if two String are anagram? Two string will be anagram if both string have same characters only the order of the character will be change in the string.

Use sorting technique to solve this problem, time complexity of this operation will be O(n log n).

1. Array rotation by k elements? Time complexity for this will be O(1).

Use below two technique to rotate the array by K

1. METHOD 1 (Using temp array)

Example:

Input arr[] = [1, 2, 3, 4, 5, 6, 7], d = 2, n =7

1) Store d elements in a temp array

temp[] = [1, 2]

2) Shift rest of the arr[]

arr[] = [3, 4, 5, 6, 7, 6, 7]

3) Store back the d elements

arr[] = [3, 4, 5, 6, 7, 1, 2]

Time complexity : O(n)

Auxiliary Space : O(d)

1. METHOD 2 (Rotate one by one)

leftRotate(arr[], d, n)

start

For i = 0 to i < d

Left rotate all elements of arr[] by one

end