Programming Construct: Intro to Recursion and Problem Solving on Recursion

Relevel
by Unacademy

Recursion

Anything which calls itself is called recursion.

Topics Covered

1. Introduction to recursion
2. Base case definition
3. Types of recursion
4. Pseudocode with working examples
5. Problem to print in increasing and decreasing order using recursion.
6. Using recursion, check if array values are sorted.
7. Fibonacci series problem
8. Factorial problem
9. Friends pairing problem.

What is recursion?

Recursion is a process of calling itself. Anything which calls itself is called a recursion process.

And, a function that calls itself is called recursive function.

How recursion works?

A recursive function calls itself. The memory for a called function is allocated on top of memory allocated to the calling function. A different copy of the local variable is created for each function call.

Factorial of a number using recursion method

We will ask the user to enter the number using a prompt.

When a user enters a negative number, an error message is thrown to enter a positive number.

When a user enters a positive number or 0 the factorial function is called.

If the number entered by the user is greater than 0, then the function will recursively call itself by decreasing the number.

This process will continue until the number becomes 1.

```
function factorial(x) {
    if(x == 1) return 1;
    return x * factorial(x-1);
}

console.log(factorial(5));
```

Fibonacci problem using recursion

```
function fib(n) {
    if(n == 0 || n == 1) return n;
    return fib(n-1) + fib(n-2);
}

console.log(fib(5));
```

Print increasing numbers using recursion

```
function f(n) {
    if(n == 1) {
        console.log(1);
        return;
    }
    f(n-1); // please print the first n-1 natural numbers for me
    console.log(n); // printing myself
}
```

Program to check if array is sorted

```
1  function issorted(arr, i) {
2      if(i == arr.length-1) return true;
3      let value = issorted(arr, i+1);
4      return value && (arr[i] <= arr[i+1]);
5  }
6
7  let arr = [5,6,4,3,2];
8  console.log(issorted(arr, 0));
9
```

Friends Pairing Problem

{1},{2},{3} : all single,
{1},{2, 3} : 2 and 3 are paired now but 1 is single,
{1, 2},{3} : 1 and 2 are paired 3 is single,
{1, 3},{2} : 1 and 3 are paired 2 is single,
Here, we need to find how many ways we can divide n elements into multiple groups.
Here group size will be max of two elements.
For n-th person there are two choices:

1. nth person remain single and we recur for f(n-1)
2. nth person pairs-up with any of the remaining (n-1) persons. We get (n-1)*f(n-2).

So, recursively we can write f(n) = f(n-1) + f(n-1) * f(n-2)

```
1  function f(n) {
2      if(n == 1 || n == 2) return n;
3      return fib(n-1) + (n-1)*fib(n-2);
4  }
5
6  console.log(f(5));
```

Assignment

1. Take an array of size 10. Calculate sum of all array elements using tail recursion.
2. Write a function that takes a string and returns if string is a palindrome.

Upcoming Teaser

1. Introduction to searching
2. Linear search
3. Binary search
4. Difference between linear search and binary search
5. Variants of binary search

Thank You