

Semana 3: Recursividad Aplicada



Datos

Materia: Estructura de Datos Aplicadas

Alumno: Eli Haziel Ortiz Ramírez

Grupo y Grado: 4 B DSM

Todos los códigos en:



Índice

Índice	2
Introducción.....	3
Actividad 3	4
Suma de Enteros Positivos (Suma Rusa):.....	4
Mínimo Común Divisor.....	5
Explicar programa	6

Introducción

La recursividad, en términos simples, es un concepto fundamental en programación que implica resolver un problema a partir de la solución de subproblemas más pequeños y similares. A través de este informe, exploraremos ejercicios prácticos que utilizan este enfoque, permitiéndonos comprender su aplicabilidad en el ámbito de la programación y sus beneficios para abordar desafíos de manera efectiva y elegante.



Actividad 3

Suma de Enteros Positivos (Suma Rusa):

Descripción: Un programa que encuentre la suma de los enteros positivos pares desde N hasta 2, Corroborar que si N es impar se imprima un mensaje de error.

```
Estructura-de-Datos - suma-rusa.ts

1  // Suma por el método Ruso usando Recursividad
2  // A   B
3  // 14  26
4  // 7   52
5  // 3   104
6  // 1   208
7  // suma = 52 + 104 + 208
8
9  const sumaRusa = (A:number,B:number, suma = 0) => {
10     let sumando = 0;
11
12     if (A ≤ 1) {
13         return suma;
14     } else if (A % 2 ≠ 0 ) {
15         Math.floor(A);
16         console.log(B);
17
18         return (sumaRusa(A/2,B * 2, suma + B));
19     } else {
20         return (sumaRusa(A/2,B * 2, suma));
21     }
22 }
23
24 console.log(sumaRusa(27,82));
25
26
27
```

Mínimo Común Divisor

Escribir un programa que calcule el máximo común divisor (MCD) de dos enteros positivos. Si $M \geq N$ una función recursiva para MCD es

$MCD = M$ si $N = \neq 0$

$MCD = MCD(N, M \bmod N)$ si $N \neq 0$

El programa le debe permitir al usuario ingresar los valores para M y N desde la consola. Una función recursiva es entonces llamada para calcular el MCD. El programa entonces imprime el valor para el MCD.

```
Estructura-de-Datos - mcd.ts

1 // Función para calcular el maximo común divisor dado 2 números
2
3 function max_comun_div(n1: number, n2: number, divisor = 2): number {
4   // Caso base: Si ambos números son 1, el max_comun_div es 1
5   if (n1 == 1 && n2 == 1) {
6     return 1;
7   } else {
8     // Verificar si ambos números son divisibles por el divisor actual divisor
9     if (n1 % divisor == 0 && n2 % divisor == 0) {
10      // Llamada recursiva dividiendo n y n1 por divisor y multiplicando el resultado por divisor
11      return divisor * max_comun_div(n1 / divisor, n2 / divisor, divisor);
12    } else if (n1 % divisor == 0) {
13      // Llamada recursiva dividiendo n por divisor
14      return max_comun_div(n1 / divisor, n2, divisor);
15    } else if (n2 % divisor == 0) {
16      // Llamada recursiva dividiendo n1 por divisor
17      return max_comun_div(n1, n2 / divisor, divisor);
18    } else {
19      // Si ninguno de los casos anteriores se cumple, incrementar divisor y llamar recursivamente
20      return max_comun_div(n1, n2, divisor + 1);
21    }
22  }
23 }
24
25 // Llamada a la función max_comun_div con los valores dados, e imprimir el resultado
26 console.log(max_comun_div(40, 60));
27
```

Explicar programa

```
Estructura-de-Datos - programa.ts

1  function f(x:number):number {
2
3      if (x > 100) {
4          console.log(x);
5          return x - 10;
6      } else {
7          console.log(x);
8          return( f( f(x+11) ) );
9      }
10 }
11 console.log(f(91));
```

El programa es una función recursiva que se repite y llama a sí misma cada que x es menor a 100, en cuyo caso siempre retorna 91, valor que equivale siempre a la cantidad de llamadas que se hace así misma hasta retornar un valor. En caso de que el valor pasado sea mayor a 100, simplemente devuelve ese número menos 10, siempre y cuando sea mayor de 100.