



Universidad Nacional Autónoma de México

Facultad de Estudios Superiores Aragón

Ingeniería en Computación

Asignatura: Estructura de datos

TAREA 11: Recursividad

Profesor: Jesús Hernández Cabrera

Alumno: Juan Diego Ortiz Cruz

Grupo: 1360

Fecha: 22/10/2024

Problema 1: sacar de un ADT pila el valor en la posición media con recursión.

Implementación:

```
1  import { PilaADT } from "./pilaADT";
2
3  function sacarPosicionMediaRecursivo(
4      pila: PilaADT<number>,
5      posicionMedia: number,
6      posicionActual: number
7  ): number | null {
8
9      if (pila.estaVacia()) {
10         throw new Error("La pila está vacía");
11     }
12
13     const elemento = pila.pop();
14
15     if (posicionActual === posicionMedia) {
16         return elemento;
17     }
18
19     const resultado = sacarPosicionMediaRecursivo(
20         pila,
21         posicionMedia,
22         posicionActual + 1
23     );
24
25     pila.push(elemento!);
26
27     return resultado;
28 }
29
30 const pila = new PilaADT<number>();
31 [1, 2, 3, 4, 5, 6, 7].forEach(num => pila.push(num));
32
33 console.log("Pila original:", pila.toString());
34 const valorMedio = sacarPosicionMediaRecursivo(
35     pila,
36     Math.floor((pila.longitud() / 2)), 0
37 );
38 console.log("Valor en la posición media:", valorMedio);
39 console.log("Pila después de la operación:", pila.toString());
```

Ejecución consola (Problema 1):

```
PS C:\Users\juani\OneDrive\Escritorio\Proyectos\Estructura-de-datos
Pila original: Pila[1, 2, 3, 4, 5, 6, 7] <- Tope
Valor en la posición media: 4
Pila después de la operación: Pila[1, 2, 3, 5, 6, 7] <- Tope
```

Problema 3: Hacer un contador regresivo con recursión.



```
1  function CuentaRegresiva(n: number): void {
2      if (n !== 0) {
3          console.log(n);
4          CuentaRegresiva(n - 1);
5      }
6  }
7
8  class Main {
9      public static main(): void {
10         CuentaRegresiva(5);
11     }
12 }
13
14 Main.main();
15
```

Ejecución consola (Problema 3)

```
PS C:\Users\juani\
5
4
3
2
1
```

Problema 4: Dado un número entero positivo, escribe un método recursivo que calcule la suma de sus dígitos

```
1 function sumaDigitos(numero: number): number {
2     // Caso base: si el número es menor que 10, retornamos el número
3     if (numero < 10) {
4         return numero;
5     }
6
7     // Caso recursivo: sumamos el último dígito con la suma de los dígitos restantes
8     return (numero % 10) + sumaDigitos(Math.floor(numero / 10));
9 }
10
11 console.log(sumaDigitos(123));
12 console.log(sumaDigitos(54321));
13
```

Ejecución consola (Problema 4)

```
PS C:\Users\juani\
6
15
```

Problema 5: Escribe un método recursivo para calcular la potencia de un número a elevado a b. a^b .

```
1  function potencia(base: number, exponente: number): number {
2      // Caso base 1: cualquier número elevado a 0 es 1
3      if (exponente === 0) {
4          return 1;
5      }
6
7      // Caso base 2: cualquier número elevado a 1 es el mismo número
8      if (exponente === 1) {
9          return base;
10     }
11
12     // Caso recursivo para exponentes positivos
13     if (exponente > 0) {
14         return base * potencia(base, exponente - 1);
15     }
16
17     // Caso recursivo para exponentes negativos
18     return 1 / potencia(base, -exponente);
19 }
20
21 console.log(potencia(2, 3)); // Salida: 8
22 console.log(potencia(5, 2)); // Salida: 25
23 console.log(potencia(3, 0)); // Salida: 1
24 console.log(potencia(4, -2)); // Salida: 0.0625
```

Ejecución consola (Problema 5)

```
PS C:\Users\juani\
8
25
1
0.0625
```