

# **UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE**

## **DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

### **ESTRUCTURA DE DATOS**



### **BÚSQUEDA BINARIA EJERCICIO 4**

**NOMBRE:** QUISHPE CHAVEZ DENISSE PAULINA

**NRC:** 29852

**FECHA:** 02/12/2025

**DOCENTE:** SOLIS ACOSTA EDGAR FERNANDO

## BÚSQUEDA BINARIA – HASH: ENCONTRAR PARES CON SUMA

Tienes N números enteros ordenados. Debes encontrar si existe un par de números cuya suma sea exactamente igual a un valor objetivo S.

Objetivo: Usar búsqueda binaria y tabla hash para encontrar el par.

Entrada:

- N: Cantidad de números
- N números ordenados
- S: Suma objetivo

Salida:

- "SI" si existe el par con los números
- "NO" si no existe

```
#include <iostream>
```

```
using namespace std;
```

```
// Variables globales usando arreglos dinamicos
```

```
long long* numeros;
```

```
int N;
```

```
// Implementacion de Busqueda Binaria
```

```
// Buscara 'objetivo' en el sub-array numeros[left...right]
```

```
bool busquedaBinaria(long long objetivo, int left, int right) {
```

```
    while (left <= right) {
```

```
        int mid = left + (right - left) / 2;
```

```
        if (numeros[mid] == objetivo) {
```

```
            return true;
```

```
        } else if (numeros[mid] < objetivo) {
```

```
            left = mid + 1;
```

```
        } else {
```

```
            right = mid - 1;
```

```
        }
```

```
}  
return false;  
}  
int main() {  
    // Lectura de N  
    if (!(cin >> N)) return 0;  
    // Asignacion de memoria dinamica  
    long long *numeros = new long long[N];  
    // Lectura de los N numeros ordenados  
    for (int i = 0; i < N; i++) {  
        if (!(cin >> numeros[i])) {  
            delete[] numeros;  
            return 0;  
        }  
    }  
    long long S;  
    // Lectura de S (Suma objetivo)  
    if (!(cin >> S)) {  
        delete[] numeros;  
        return 0;  
    }  
    bool encontrado = false;  
    long long num1 = 0, num2 = 0;  
    // Busqueda del par usando Busqueda Binaria (O(N log N))  
    for (int i = 0; i < N; i++) {  
        long long complemento = S - numeros[i];  
        // Se busca el complemento solo si es mayor que el numero actual  
        // y en el sub-array [i + 1, N - 1] para evitar duplicados.
```

```
if (complemento > numeros[i]) {  
    if (busquedaBinaria(complemento, i + 1, N - 1)) {  
        encontrado = true;  
        num1 = numeros[i];  
        num2 = complemento;  
        break;  
    }  
}  
}  
  
// Salida Requerida  
if (encontrado) {  
    cout << "SI\n";  
    cout << num1 << " " << num2 << "\n";  
} else {  
    cout << "NO\n";  
}  
  
// Liberacion de memoria dinamica  
delete[] numeros;  
return 0;  
}
```