

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

ESTRUCTURA DE DATOS



BÚSQUEDA BINARIA EJERCICIO 1

NOMBRE: QUISHPE CHAVEZ DENISSE PAULINA

NRC: 29852

FECHA: 02/12/2025

DOCENTE: SOLIS ACOSTA EDGAR FERNANDO

ORGANIZADOR DE LIBROS EN BIBLIOTECA

Tienes N libros con diferentes grosores (en cm) y K estantes disponibles. Los libros deben colocarse en el orden dado (no puedes reordenarlos).

Objetivo: Encontrar el ancho mínimo que deben tener los estantes para que todos los libros quepan.

Entrada

- N: Número de libros
- K: Número de estantes disponibles
- Grosores: El grosor de cada libro en centímetros

Salida: El ancho mínimo (en cm) que deben tener los estantes.

```
#include <iostream>
using namespace std;
long long* grosores;
int N, K;
// Verifica si los libros caben en K estantes con ancho W
bool puedeCaber(long long W) {
    int estantes = 1;
    long long actual = 0;
    for (int i = 0; i < N; i++) {
        if (grosores[i] > W) return false;
        if (actual + grosores[i] <= W) {
            actual += grosores[i];
        } else {
            estantes++;
            actual = grosores[i];
        }
    }
    return estantes <= K;
}
```

```
        if (estantes > K) return false;
    }
}
return true;
}
```

```
int main() {
    cout << "Libros: ";
    cin >> N;
    cout << "Estantes: ";
    cin >> K;
    grosos = new long long[N];
    long long suma = 0, maximo = 0;
    cout << "Grosos (cm): ";
    for (int i = 0; i < N; i++) {
        cin >> grosos[i];
        suma += grosos[i];
        if (grosos[i] > maximo) maximo = grosos[i];
    }
    // BÚSQUEDA BINARIA
    // Busca el ancho mínimo entre [maximo, suma]
    long long left = maximo; // Mínimo: el libro más grueso
    long long right = suma; // Máximo: todos los libros juntos
    long long respuesta = suma;
    while (left <= right) {
        long long mid = left + (right - left) / 2;
        if (puedeCaber(mid)) {
            respuesta = mid;
            right = mid - 1; // Intentar con menos
        } else {
```

```
    left = mid + 1; // Necesitamos más
}
}
cout << "\nAncho minimo: " << respuesta << " cm\n";
// Mostrar distribución
cout << "\nDistribucion:\n";
int est = 1;
long long ancho = 0;
cout << "Estante " << est << ": ";
for (int i = 0; i < N; i++) {
    if (ancho + grosores[i] <= respuesta) {
        cout << grosores[i] << " ";
        ancho += grosores[i];
    } else {
        cout << "(" << ancho << "cm)\n";
        est++;
        cout << "Estante " << est << ": " << grosores[i] << " ";
        ancho = grosores[i];
    }
}
cout << "(" << ancho << "cm)\n";
delete[] grosores;
return 0;
}
```