

# TEAM REFERENCE DE LOS SURFISTAS BOLIVIANOS

Universidad Mayor de San Simón

**Contents**

|

# 1 Sección de ejemplo

## 1.1 Código de ejemplo

**Descripción:** Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

**Complejidad Temporal:**  $O(n\log(n) + m^2)$

**Usos:** Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

```
#include <iostream>

int main() {
    std::cout << "Hola_Mundo" << std::endl;
    return 0;
}
```

## 2 Estructuras de datos

### 2.1 SegmentTree

**Descripción:** Estructura de datos que almacena un arreglo de elementos siguiendo la lógica de un árbol binario dividiendo el arreglo original en subintervalos. Gracias a la forma de almacenar los datos, un SegmentTree es capaz de realizar cálculos y actualizaciones en intervalos de la forma  $[l, r]$  eficientemente.

**Complejidad Temporal:**  $O(\log(n))$

**Usos:** Sumas de elementos en intervalos, restas de elementos en intervalos, máximo elemento de un intervalo, mínimo elemento de un intervalo, etc.

```
/*
Para construir un nuevo SegmentTree se tienen que tomar
en cuenta dos cosas: el valor que se almacenará en cada
nodo y la operación encargada de fusionar los valores.
*/
void build(int a[], int v, int tl, int tr) {
    if (tl == tr) {
        t[v] = a[tl];
    } else {
        int tm = (tl + tr) / 2;
        build(a, v*2, tl, tm);
        build(a, v*2+1, tm+1, tr);
        t[v] = t[v*2] + t[v*2+1];
    }
}
```