



# Algorithm

`#include <algorithm>`



# Funciones lambda

- Te permiten personalizar las funciones de la librería *algorithm*.



```
1  auto cmp = [&externo](tipo a, tipo b) {  
2      return a > b; // booleano  
3  };
```

# find


find(begin, end, valor)



```
1  vector<int> v = { 3, 5, 4, 6, 10 };
2
3  auto it = find(v.begin(), v.end(), valor);
4
5  if (it != v.end()) {
6      cout << "Found " << *it << endl;
7  } else {
8      cout << "Not found" << endl;
9  }
```

# find\_if

find\_if(begin, end, cmp)



```
1  vector<int> v = { 3, 5, 4, 6, 10 };
2
3  auto cmp = [&n](int a) {
4      return a > n;
5  };
6
7  auto it = find_if(v.begin(), v.end(), cmp);
8
9  if (it != v.end()) {
10     cout << "Found " << *it << endl;
11 } else {
12     cout << "Not found" << endl;
13 }
```

# sort

sort(begin, end, cmp)



```
1  vector<int> v = { 3, 5, 4, 6, 10 };
2
3  auto cmp = [](int a, int b) {
4      return a > b;
5  };
6
7  sort(v.begin(), v.end(), cmp);
```

---