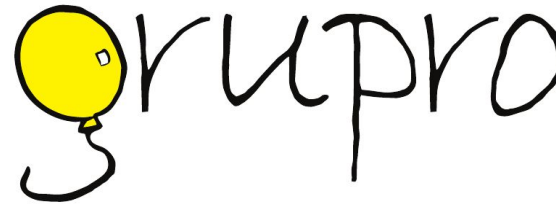




Curso de Programação Nível Intermediário



Universidade Federal da Bahia
Instituto de Computação
Departamento de Ciência da Computação

AULA 2 - ORDENAÇÃO (VECTOR + STRUCT)

Ordenação - sort

```
#include <iostream>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
using namespace std; // sort -> std
```

```
int main() {
```

```
    vector<int> v;
```

```
    for(int i=0; i < 1000; i++) {
```

```
        int j; cin >> j;
```

```
        v.push_back(j);
```

```
    }
```

```
// ordenação com complexidade ~ NlogN (não estável)
```

```
sort (v.begin(), v.end());
```

```
}
```

Saída:

5 2 7 1 7

1 2 5 7 7

Ordenação - stable_sort

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std; // sort -> std
int main() {
    vector<int> v;
    for(int i=0; i < 1000; i++) {
        int j; cin >> j;
        v.push_back(j);
    }
    stable_sort (v.begin(), v.end()); // ordenação com complexidade = NlogN
}
```

Qd usado em struct, se não definir o critério de ordenação de todos os campos, ao empatar em campos definidos na função de comparação, a ordenação mantém a ordem da entrada, portanto, estável

Saída:

```
5 2 7 1 7
1 2 5 7 7
```

Ordenação - reverse

```
#include <iostream>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
using namespace std; // sort -> std
```

```
int main() {
```

```
    vector<int> v;
```

```
    for(int i=0; i < 1000; i++) {
```

```
        int j; cin >> j;
```

```
        v.push_back(j);
```

```
    }
```

```
    reverse(v.begin(), v.end()); // ordenação em ordem decrescente
```

```
}
```

Saída:
5 2 7 1 7
7 7 5 2 1

Ordenação - sort & struct

```
struct pessoa {  
    int id;  
    string nome;  
};  
  
bool cmp(pessoa i, pessoa j) {  
    return ( i.id < j.id  ||  i.id == j.id  &&  i.nome < j.nome);  
}  
  
int main() {  
    vector<pessoa> v;  //...  
    stable_sort (v.begin(), v.end(), cmp);  
}
```

Ordenação - sort & struct

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <string>

using namespace std;

struct pessoa {
    int id;
    string nome;
};

bool cmp(pessoa i, pessoa j) {
    return ( i.id < j.id || i.id == j.id && i.nome < j.nome);
}
// continua ...
```

```
int main() {
    vector<pessoa> v;
    pessoa x;

    for (int i=0; i<4 ; i++){
        cin >> x.id >> x.nome;
        v.push_back(x);
    }
    // vetor sem ordenação
    for (int i=0; i<4 ; i++)
        cout << v[i].id << " " << v[i].nome << endl;
    cout << endl;

    stable_sort (v.begin(), v.end(), cmp);

    cout << "Vetor ordenado" << endl;
    for (int i=0; i<4 ; i++) // vetor ordenado
        cout << v[i].id << " " << v[i].nome << endl;
}
```

Saída:
5 Jose
3 Maria
6 Carlos
6 Ana
Vetor ordenado
3 Maria
5 Jose
6 Ana
6 Carlos

Vetores - sort

Saiba mais em:

<http://www.cplusplus.com/reference/algorithm/sort/>