

# Resumo STL

September 8, 2018

## 1 Containers Importantes

Os mais usados são **vector**, **pair**, **map**, **set**.

### 1.1 Vector

Refinamento das classes Random Access Sequence e Back Insertion Sequence, se comporta como um array. Definido em `<vector>`.

Funções definidas:

- `empty`, `size`, `clear`, `push_back`, `pop_back`, `insert`, `erase`, `begin`, `end`

```
vector<int>::iterator cur = vet.begin() \\Usando iterator
vector<int>::iterator end = vet.end() \\Usando iterator
while (; cur != end, ++cur)
    cout << *cur << endl;
```

### 1.2 Bitset

### 1.3 Linked List

### 1.4 Stack

### 1.5 Queue and Double Queue

### 1.6 Pair

Pair é uma struct, e não uma classe.

Pra fazer um pair:

```
make_pair(first, second)
```

## 1.7 Balanced Search Tree (Map/Set)

Tabela de símbolos genérica, com chave e valor.

Definido em **<map>**.

É possível iterar sobre o mapa, usando begin() e end().

Funções definidas:

- insert(pair<key, value>):pair<iterator, bool>, find(key):iterator, begin():iterator, end():iterator

```
map<string , vector<int> > teste ;  
vector<int> numeros(10, 0);
```

```
pair<map<string , vector<int> >::iterator , bool> resultado ;  
resultado = ex.insert(make_pair(" lucas ", numeros));
```

## 1.8 Heap (priority\_queue)

## 1.9 Hash Table (Unordered\_map)

## 1.10 Graph

# 2 Algoritmos Importantes

- find(first, last, value):
- copy(first, last, result):
- swap(a, b):
- remove(first, last, value):
- unique(first, last [,pred]):
- reverse(first, last):
- sort(first, last [, comp]):
- stable\_sort(first, last [,comp]):
- merge(first1, last1, first2, last2, result [,comp]):
- inplace\_merge():
- includes():
- set\_union():
- is\_sorted():
- min(), max():
- iota():

- `binary_search()`;
- `next_permutation()`;
- `min_element()`;
- `max_element()`;

### 3 Misc

```
#define _foreach(it, b, e) for (typeof(b) it = (b); it != (e); it++)
#define foreach(x...) _foreach(x)
#define all(x) (x).begin(), (x).end()
#define UNIQUE(c) c.resize(unique(all(c)) - (c).begin())

#define INF 1000000000

typedef long long ll
typedef pair<int, int> ii
typedef pair<int, ii> iii
typedef vector<int> vi
typedef vector<ii> vii

//memset(arr, 0, sizeof arr); // Clear array

// Imprimir N casas de Pi
double pi = 2 * acos(0,0);
cin << n << pi;
printf("%. *lf\n", n, pi);

// Imprimir numeros distintos ordenados
sort(ALL(v)); UNIQUE(v);

// Gen todos os subsets
int p[20], N = 2
for(i = 0; i < 20; i++) p[i] = i
for(i = 0; i < (1 << N); i++)
    for(j = 0; j < N; j++)
        if (i & (1 << j))
            printf("%d ", p[j])
        printf("\n")
```