

Standard Template Library

Tópicos Avançados de Programação

Instituto de Informática

1 de outubro de 2017

STL

String

pair

Set

Map

Vector

Queue

Stack

Priority_queue

- string
- pair
- set
- map
- vector
- queue
- stack
- priority_queue

STL

String

pair

Set

Map

Vector

Queue

Stack

Priority_queue

String

String

STL

String

pair

Set

Map

Vector

Queue

Stack

Priority_queue

O que é?

Estrutura de dados similar ao vetor de char.

Exemplo de string

```
#include <bits/stdc++.h>
using namespace std;
int main( int argc , char **argv )
{
    string s = "abcdefg";
    string t = "hijklmno"
    cout << s[0] << '\n';
    cout << s.size() << '\n';
    s += t;
    cout << s << '\n';
    return 0;
}
```

STL

String

pair

Set

Map

Vector

Queue

Stack

Priority_queue

pair

pair

STL

String

pair

Set

Map

Vector

Queue

Stack

Priority_queue

O que é?

Estrutura de dados que armazena uma par de dados ou de estruturas de dados.

Exemplo de pair

```
#include <bits/stdc++.h>
using namespace std;
int main( int argc , char **argv )
{
    pair<int,int> p;
    p = make_pair(0,1);
    cout << p.first << ' ' << p.second << '\n';
    return 0;
}
```


STL

String

pair

Set

Map

Vector

Queue

Stack

Priority_queue

Set

Set

STL

String

pair

Set

Map

Vector

Queue

Stack

Priority_queue

O que é?

Estrutura de dados que mantém seus elementos ordenados e todos os elementos são distintos.

Exemplo de set

```
#include <bits/stdc++.h>
using namespace std;
int main( int argc , char **argv )
{
    set<int>s;
    set<int>::iterator it;
    s.insert(2);
    it = s.find(2);
    s.begin(); s.end();
    cout << s.size() << '\n';
    return 0;
}
```

STL

String

pair

Set

Map

Vector

Queue

Stack

Priority_queue

Map

Map

STL

String

pair

Set

Map

Vector

Queue

Stack

Priority_queue

O que é?

Estrutura de dados que mantém seus elementos ordenados pela chave e não pelo valor. Parecido com hash.

Exemplo de map

```
#include <bits/stdc++.h>
using namespace std;
int main( int argc , char **argv )
{
    map<int , int> mapa;
    mapa[2] = 3;
    map<string , int> mapa_string;
    mapa_string["oi"] = 5;
    cout << mapa["oi"] << '\n';
    return 0;
}
```

STL

String

pair

Set

Map

Vector

Queue

Stack

Priority_queue

Vector

Vector

STL

String

pair

Set

Map

Vector

Queue

Stack

Priority_queue

O que é?

Estrutura de dados similar a uma lista encadeada, porém com a mesma facilidade de acesso de um array.

Exemplo de vector

```
#include <bits/stdc++.h>
using namespace std;
int main( int argc , char **argv )
{
    vector<int>v;
    v.push_back(2);
    sort(v.begin(),v.end());
    cout << v.size() << ' ' << v[0] << '\n';
    return 0;
}
```

STL

String

pair

Set

Map

Vector

Queue

Stack

Priority_queue

Queue

Queue

STL

String

pair

Set

Map

Vector

Queue

Stack

Priority_queue

O que é?

Estrutura de dados similar a uma fila, onde cada elemento é inserido no final da fila e retirado do começo da fila.

Exemplo de queue

```
#include <bits/stdc++.h>
using namespace std;
int main( int argc , char **argv )
{
    queue<int>q;
    q.push(1);
    cout << q.front() << '\n';
    q.pop();
    cout << q.size() << '\n';
    return 0;
}
```

STL

String

pair

Set

Map

Vector

Queue

Stack

Priority_queue

Stack

Stack

STL

String

pair

Set

Map

Vector

Queue

Stack

Priority_queue

O que é?

Estrutura de dados similar a uma pilha, onde cada elemento é inserido no começo da pilha e retirado do início da pilha.

Exemplo de stack

```
#include <bits/stdc++.h>
using namespace std;
int main( int argc , char **argv )
{
    stack<int>st;
    st.push(1);
    cout << st.top() << '\n';
    st.pop();
    cout << st.size() << '\n';
    return 0;
}
```

STL

String

pair

Set

Map

Vector

Queue

Stack

Priority_queue

Priority_queue

Priority_queue

STL

String

pair

Set

Map

Vector

Queue

Stack

Priority_queue

O que é?

Estrutura de dados similar a uma fila, porém com prioridade. O maior elemento fica no início da priority_queue. Similar a uma heap.

Exemplo de priority_queue

```
#include <bits/stdc++.h>
using namespace std;
int main( int argc, char **argv )
{
    priority_queue<int>pq;
    pq.push(1);
    cout << pq.top() << '\n';
    pq.pop();
    cout << pq.size() << '\n';
    return 0;
}
```