

Some notes on STL

Algorithms and Data Structures

ESCI

<http://www.cplusplus.com/reference/list/list>

Constructors:

- `list<T>()`: constructs an empty list.
- `list<T>(int size, const T& value)`: constructs a list with `n` elements, each of which is a copy of `value`.

Modifiers:

- `.push_back(T)`, `.pop_back()`
- `.push_front(T)`, `.pop_front()`

Iterators:

- `.begin()`, `.end()`: returns iterator
- `.rbegin()`, `.rend()`: returns reverse_iterator

Capacity:

- `.empty()`
- `.size()`

<http://www.cplusplus.com/reference/stack/stack/>

Constructor:

- `stack<T>()`: constructs an empty stack.

Access:

- `.top()`

Modifiers:

- `.push(T)`
- `.pop()`

Capacity:

- `.empty()`
- `.size()`

<http://www.cplusplus.com/reference/map/map>

Constructor:

- `map<Key, T>()`: constructs an empty map (increasing key ordering).

Operations:

- `.find(Key)`: returns an iterator to the element (if it exists), or `.end()`.

Modifiers:

- `[Key]`: it always increases the size of the map by 1.
- `.erase()`: by position, key or range.

Iterators:

- `.begin()`, `.end()`: returns iterator
- `.rbegin()`, `.rend()`: returns reverse_iterator

Capacity:

- `.empty()`, `.size()`

<http://www.cplusplus.com/reference/queue/queue>

Constructor:

- `queue<T>()`: constructs an empty queue.

Access:

- `.front()`: returns the first element.
- `.back()`: returns the last element.

Modifiers:

- `.push(T)`: adds T value at the end of the queue.
- `.pop()`

Capacity:

- `.empty()`
- `.size()`

priority queue

http://www.cplusplus.com/reference/queue/priority_queue/

Constructor:

- `priority_queue<T>()`: constructs an empty priority queue.

Access:

- `.top()`: returns the greatest element.

Modifiers:

- `.push(T)`
- `.pop()`

Capacity:

- `.empty()`
- `.size()`