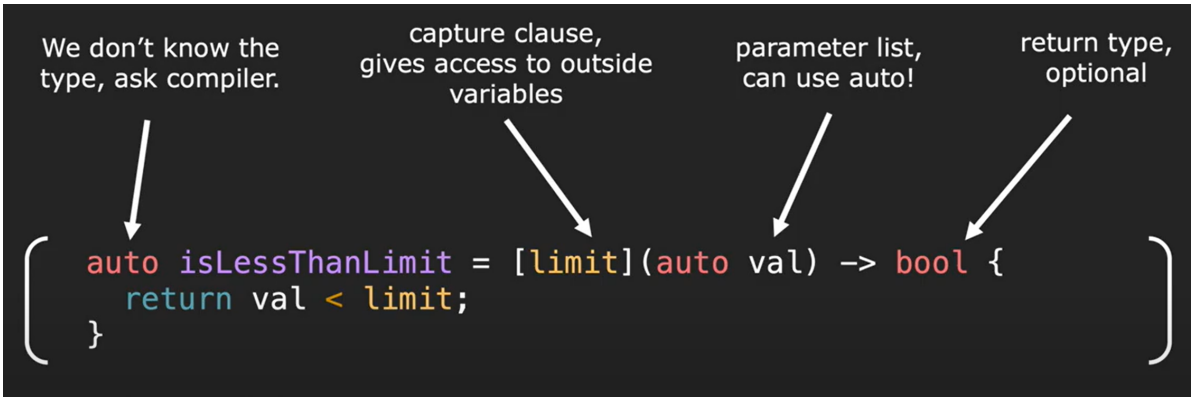
STL sort实现原理




```cpp
// capture all by value, except teas is by reference
auto func1 = [=, &teas](parameters) -> return-value {
    // body
};

// capture all by reference, except banned is by value
auto func2 = [&, banned](parameters) -> return-value {
    // body
};
```

```cpp
template <typename InputIterator, typename fun>
int MyAgrthmFun(InputIterator start, InputIterator end, fun elemToCount)
{

    int counter = 0;
    InputIterator iter;
    for (iter = start; iter != end; ++iter)
    {
        if (elemToCount(*iter))
            ++counter;
    }
    return counter;
}

int main()
{

    std::vector<int> v{3, 1, 4, 1, 5, 9, 3};
    std::list<int> l{2, 7, 1, 8, 2, 8, 2};

    int limit=5;
    auto IsLessThan=[limit](auto val){
        return val<limit;
    };

    cout << "vec count : " << MyAgrthmFun(v.begin(), v.end(), IsLessThan) << endl;

    cout << "list count : " << MyAgrthmFun(l.begin(), l.end(), IsLessThan) << endl;
}
```

```cpp
#include <iostream>
#include <sstream>
#include <vector>
#include <deque>
#include <iterator>
#include <list>
#include <set>
#include <algorithm>
using namespace std;

template <typename InputIterator, typename fun>
int MyAgrthmFun(InputIterator start, InputIterator end, fun elemToCount)
{

    int counter = 0;
    InputIterator iter;
    for (iter = start; iter != end; ++iter)
    {
        if (elemToCount(*iter))
            ++counter;
    }
    return counter;
}

int main()
{

    std::vector<int> v{3, 1, 4, 1, 5, 9, 3};
    std::list<int> l{2, 7, 1, 8, 2, 8, 2};

    int limit = 5;
    auto IsLessThan = [limit](auto val)
    { return val < limit; };

    // sort()的实现方法
    cout << "vec count : " << MyAgrthmFun(v.begin(), v.end(), IsLessThan) <<
endl;
    cout << "list count : " << MyAgrthmFun(l.begin(), l.end(), IsLessThan) <<
endl;

    // stl输出方法，通过copy函数拷贝到ostreamiterator进行输出
    copy(v.begin(), v.end(), ostream_iterator<int>(cout, " "));
    cout << endl;

    // 检测容器内符合条件的元素并swap到末尾
    cout << "v before remove : ";
    copy(v.begin(), v.end(), ostream_iterator<int>(cout, " "));
    cout << endl;

    v.erase(remove_if(v.begin(), v.end(), IsLessThan), v.end());

    cout << "v after remove : ";
    copy(v.begin(), v.end(), ostream_iterator<int>(cout, " "));
    cout << endl;
```

```
    // copy 符合条件的元素到新vector
    v = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
    vector<int> to_vector;

    copy_if(v.begin(), v.end(), back_inserter(to_vector), IsLessThan);

    cout << "to_vector : ";
    copy(to_vector.begin(), to_vector.end(), ostream_iterator<int>(cout, " "));
    cout << endl;
}
```