

# STL sort

Murashko Ivan

## Introduction

The most tricky thing in C++ is an error catch. Most part of errors are located in the application level and you can be quite sure that the library, especially the well known STL, is error free. But what should you do if your application code is trivial i.e. seems to be error free but you got a SIGSEGV inside a system library. Most probably you just felt into a situation that is described in the article.

## 1 Example

Lets look at the following code:

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

void print(std::vector<int>& data) {
    for (int item : data) {
        std::cout << item << " ";
    }
    std::cout << std::endl;
}

int main(){
    std::vector<int> data = {1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                            1, 1, 1, 1, 1, 1, 1, 1};
    auto comp = [](int i1, int i2) { return i1 <= i2; };
    std::sort(data.begin(), data.end(), comp);
    print(data);
    return 0;
}
```

What's output does it produces? One can expect the following one:

```
$
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$
```

i.e. the original array has to be displayed. In reality the following output is produced by gcc 5.5.0 <sup>1</sup>

```
$  
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
$
```

As one can see the array is broken. How it can be if every function in the code snapshot seems to be correct?

[1]

## References

- [1] Wikipedia contributors. Quicksort — Wikipedia, the free encyclopedia. — <https://en.wikipedia.org/w/index.php?title=Quicksort&oldid=917319740>. — 2019. — [Online; accessed 23-September-2019].

---

<sup>1</sup> I used gcc:5 docker image for the compilation