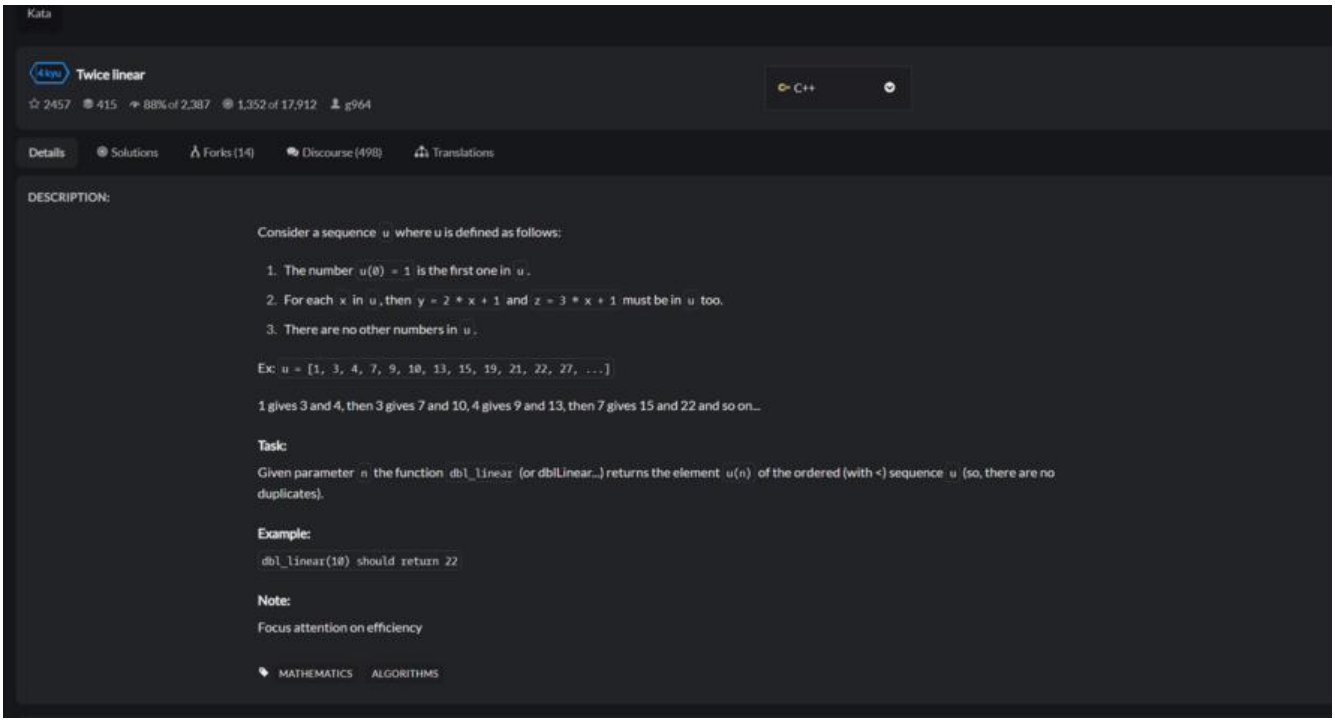
链滴

【KATA 练习日记】关于 std::set 的使用

作者：skjsnb
原文链接：https://ld246.com/article/1675613214292
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)



0. 前言

今天练习的题目是“Twice linear”，题目的概述如下：

Consider a sequence u where u is defined as follows:

1. The number u(0) = 1 is the first one in u.
2. For each x in u, then y = 2 * x + 1 and z = 3 * x + 1 must be in u too.
3. There are no other numbers in u.

Ex: u = [1, 3, 4, 7, 9, 10, 13, 15, 19, 21, 22, 27, ...]

1 gives 3 and 4, then 3 gives 7 and 10, 4 gives 9 and 13, then 7 gives 15 and 22 and so on...

Task:

Given parameter n the function dbl_linear (or dblLinear...) returns the element u(n) of the o dered (with <) sequence u (so, there are no duplicates).

Example:

dbl_linear(10) should return 22

Note:

Focus attention on efficiency

由于限制运算时间，这道题最终我卡在了容器内元素的排序和去重这一关。。。

看了下大佬的标准答案，发现他们都使用了 std::set 这一种容器。

```
#include <set>

class DoubleLinear
{
public:
    static int dblLinear(int n)
    {
      std::set<int>  seq;

      seq.insert(1);
      std::set<int>::iterator it = seq.begin();
      for (int i = 0; i < n; ++it, ++i)
      {
        int current = *it;
        seq.insert(2 * current + 1);
        seq.insert(3 * current + 1);
      }

      return *it;
    }
};
```

---

1. std::set 容器简介

- std::set 容器是一种按照特定顺序存储的容器。
- 在 std::set 容器内，元素的值即为其的键，并且每个值必须是唯一的。
- 对于 std::set 容器，其内部元素的值是不可被修改的，但可以插入元素和删除元素。
- 在 std::set 容器内，元素始终按照指定的 **比较方式**进行排序，默认是从小到大。

2. std::set 容器的使用方法

2.1 demo: set 容器特性

```
#include <iostream>
#include <set>

void print_set(std::set<int>& buffer){
    for (auto& item:buffer) {
        std::cout << item << " ";
    }
    std::cout << "\r\n";
}

void foo1(){
    std::set<int> buf;
    buf.insert(10);
    buf.insert(5);
```

```
    buf.insert(10);
    print_set(buf);
}

int main() {
    foo1();
    return 0;
}
```

程序输出结果：

```
5 10
```

这个 demo 主要检查了 set 容器的顺序存储和元素唯一的特性，buf 容器顺序插入了三个元素，分是 {10,5,10}，从打印结果看出 set 容器默认的排序方式为从小到大，且元素唯一。

2.2 demo: set 容器的遍历

```
void traversal_set(std::set<int>& buffer){
    // 方式1：使用基于范围的for循环
    for (auto& item:buffer) {
        std::cout << item << " ";
    }
    std::cout << "\r\n";
    // 方式2：使用迭代器
    auto it = buffer.begin();
    while (it != buffer.end()){
        std::cout << *it << " ";
        it++;
    }
    std::cout << "\r\n";
}
```

2.3 demo: set 容器的排序方式

```
void foo3(){
    auto cmp = [](int a,int b){return a > b;};
    std::set<int, decltype(cmp)> buf(cmp);
    buf.insert(10);
    buf.insert(5);
    buf.insert(10);

    for (auto& item:buf) {
        std::cout << item << " ";
    }
    std::cout << "\r\n";
}
```

这个 demo 使用了 lambda 函数定义了容器的排序方式，最终输出结果为：

```
10 5
```

关于 set 容器的排序方式，还有其他写法，详情可查看 c++ - Using custom std::set comparator - S

ack Overflow 高赞回答。

2.4 demo: set 容器的元素插入和删除

```
void foo4(){
    std::set<int> buf;
    buf.insert(10);
    buf.insert(5);
    buf.insert(8);
    buf.erase(5);
    print_set(buf);
}
```

程序输出：

```
8 10
```

---

3. 参考资料

- https://cplusplus.com/reference/set/set/
- c++ - Using custom std::set comparator - Stack Overflow
- std::set - C++ 中文 - API 参考文档 (apiref.com)

---

4. 代码仓库

- kata/main.cpp at main · skjsnb/kata (github.com)
- my_cpp_learn/main.cpp at main · skjsnb/my_cpp_learn (github.com)