

注意：本学期 CS101 PA 的所有 C++ 代码都采用 C++20 标准。如果您使用 GCC 或 Clang，您需要在编译时设置 `-std=c++20`。如果您看到类似这样的报错

```
g++-9: error: unrecognized command line option '-std=c++20'; did you mean '-std=c++2a'?
```

说明您的编译器版本过低，请安装更高版本的编译器。目前 GCC 13 几乎已经将 C++20 除 modules 外的全部新特性实现完毕，我们推荐使用 GCC 13。

Quick Sort

In this problem, you are required to implement a quick sort algorithm.

Recall: The algorithm selects a pivot element from the array and partitions the other entries into two sub-arrays, according to whether they are less than or greater than the pivot. The sub-arrays are then recursively sorted using the same process until the entire array is sorted.

The files attached to this problem are shown as follows. We provide a test to help you check your code.

```
.
├─ attachments/
│   └─ tests/
│       └─ test_sorting.cpp
└─ quick_sort.hpp
```

NOTE:

1. You should **NEVER** call the copy constructor of data type in the whole algorithm.
2. You should **NEVER** call `std::sort()` or use other algorithms to complete sort in the whole algorithm.
3. In the algorithm, you should keep partitioning the whole array with pivots until there is **no more than one entry in the part**.

In the template, we have provided a function named `random_choice`. You should pass two *Random Access Iterators* to them. The return value will be an iterator pointing to a randomly chosen entry.

Here is an example of usage:

```
int main() {
    std::vector a{1, 4, 2, 8, 5, 7};

    std::cout << *random_choice(a.begin(), a.end()) << std::endl;
    std::cout << *random_choice(a.begin(), a.end()) << std::endl;
    std::cout << *random_choice(a.begin(), a.end()) << std::endl;

    return 0;
}
```

A possible output could be:

4
4
7

Requirements:

Your task is to complete the function `void quick_sort(std::random_access_iterator auto begin, std::random_access_iterator auto end, auto compare)`. About the usage of `compare`, you could refer to the description "特殊操作" of `prob1: Slist`.

After sorting, you should guarantee that for every two iterators `x, y` $\in [begin, end)$, whenever `compare(*x, *y)` returns `true`, we have `x < y`.

提交与评分

在 OJ 上提交 `quick_sort.hpp` 内的全部内容。

本题的评分由 OJ 分数（60%）和线下 check（40%）两部分构成。线下 check 会在此次作业结束时间之后进行。

注：线下 check 也带有检查学术诚信的含义，当然这不是唯一的手段。如果被认定为抄袭，OJ 的分数也会作废，并且会有惩罚。**特别强调，抄袭来自 generative AI 的代码和抄袭网上的代码是同等处理的，我们建议您在写作业时关闭一切 generative AI 工具。**