

编程任务：

1. 将数组分为两部分

实现以下声明的分区方法，将给定数组 A 分为两部分。 j ($j \geq 0$) 是枢轴的索引，将所有元素 $<$ 枢轴向左，元素 \geq 枢轴向右。

```
1 模板<类型名 T>
2  int Partition (数组<T> A, int
3  j, ... ) {
4  }
```

2. 非随机 Quicksort

实现快速排序算法 Quicksort，并以数组中的最后一个元素为支点。使用上面定义的分区间作为其子例程。

```
1 模板<类型名 T>
2  void Quicksort (数组<T>
3  A, ... ) {
4  }
```

3. 随机选择算法

实施方法 Rselect，以找到给定数组 A 的第 i 个 ($i > 0$) 最小项。从 A 随机选择均匀的轴。伪代码如下。

```
1  template <类型名 T>
2  TR select (数组<T> A, int i, ...) {
3      如果 A 中只有一个元素返回 A [0];
4      从数组 A 中随机地均匀选择枢轴; 使用 Pivotp 进行分配 A;
5      令 j 为 p 的索引;
6
7      if (j == i) 返回 p;
8      if (j > i) 返回 R select (A 的第一部分,
9      其他 i, ...) ;
10 }
```

4. 确定性选择算法

使用确定性选择算法实现方法 Dselect，以找到数组 A 的第 i 个 ($i > 0$) 最小项。通过以下步骤选择枢轴

- (a) 将 A 分成 $n / 5$ 组，每组 5 个
- (b) 对每个组进行排序（例如，使用插入排序，合并排序...）
- (c) 将 $n / 5$ 个中值复制到新数组 C 中

- (d) 通过调用确定性选择算法来递归计算 C 的中位数
- (e) 返回 C 的中位数作为枢轴

Dselect 的伪代码如下

```

1  template <类型名 T>
2  TD 选择 (数组<T> A, int i,      ....) {
3      if (A 中只有一个元素) 返回 A [0];
4      / ChoosePivot
5      将 A 分成 5 组, 每组排序; C = n / 5 个中位数;
6      p = D 选择 (C, n / 10, ... ) ; //返回 C 中的中值
7
8
9      // 划分
10     使用 Pivotp 进行分配 A; 令 j 为
11     p 的索引;
12     如   = i) 返回 p;
13     果   > i) 返回 R select (A, i 的第一部分 ,...);
14     , : 返回 R select (A 的第二部分, i - j -
15 }
```

5. 完整的主要功能

main.cpp 是我们提供的, 请尝试按照其中的说明来测试您的功能。

重要笔记:

- 参考: 第 7 课的课堂笔记。
- 如果输入数组为空, 请确保您的程序不会崩溃。
- 记住提交您的 makefile!
- 到期日: 2019/10/26 11:59 pm