人体快排计算机技术文档暨实验报告

计算机科学与技术导论 A310 - 4

1. 人体计算机的组成

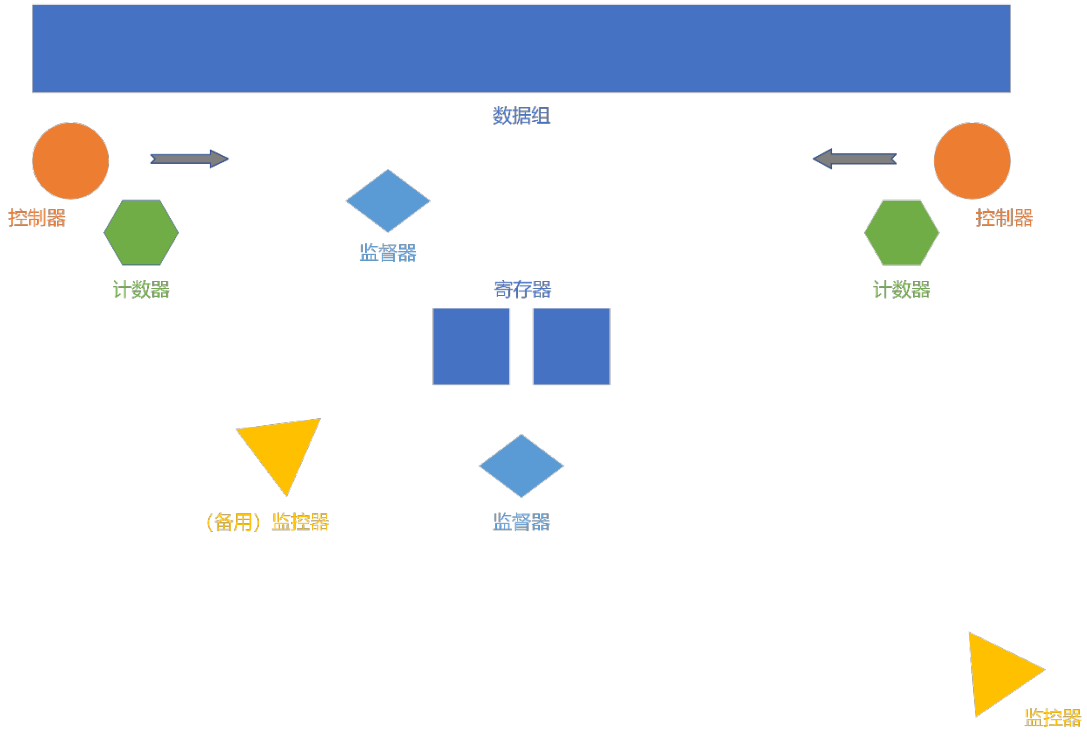
1.1. 成员列表

姓名	承担任务
肖予纯	
陈一韬	
操淳	
陈利通	
陈中浩	
崔洛雯	控制器
崔涛	
戴琪智	
邓堯	
杜培绪	

姓名	承担任务
杜雪怡	
方乾瑶	
方欣	
费廷哲	
冯敏	
高笑颜	
龚奕霖	
郭千纯	
何雨琦	
和思凯	

姓名	承担任务
洪丽杰	
洪子翔	
袁鸿东	计数器
翟一航	监督器
黄羽轩	计数器
王佳晖	监督器
万展廷	监控器
于景一	控制器
袁瑞彬	监控器

1.2. 场地设置及解释



① 寄存器

为出列的数据（基准值和数据）提供固定的区域，在此处方便比较身高。

② 数据组

排序前，数据组按学号顺序站成一列。由控制器控制出、入列。
排序好的分区可以坐下休息。

③ 控制器与计数器

控制器通过指令完成快排过程。当且仅当将运行指令时，计数器大声报数，控制器再执行。单个控制器只统计同侧的控制器执行的指令数。

④ 监督器与监控器

监督器避免程序错误，一个监督出、入列操作，另一个监督身高比较过程。
监控器负责录制视频、拍摄瞬间。

1.3. 特别说明

快速排序的基本过程需要分别从左端和右端对面，逐一进行判断。

我们清楚，为确保程序是串行执行、并拟合计算机的基本要求，只能同时有一个控制器在运作，相应的，实验中安排的另一个控制器必须为“备份控制器”。

依照这个思路，我们规定：

① 控制器

要求**两个控制器***分别*在两端，依次进行这个过程。

当**左边的控制器**在运行时，**右边的控制器**就为*“备份控制器”*，是并不在工作的，直到左侧进行完，才将左、右控制器的状态互换，此时**右边的控制器**在运行，**左边的控制器**为*“备份控制器”*。按照这个规律运行程序。

为便于编写代码，控制器名称只与初始状态有关。一开始在左边的控制器就永远叫“左边”，右边同理。

② 计数器

计数器也同样需要遵循这个规则。当左边的控制器在运行，左侧的计数器便运作，此时右侧的不在工作，就是*“备份计数器”。*

特别的，为避免重复计数，规定**“互换位置”、“基准值入列”**的指令由**特定的控制器**计数。

2. 指令集与代码

2.1. 概述与思路

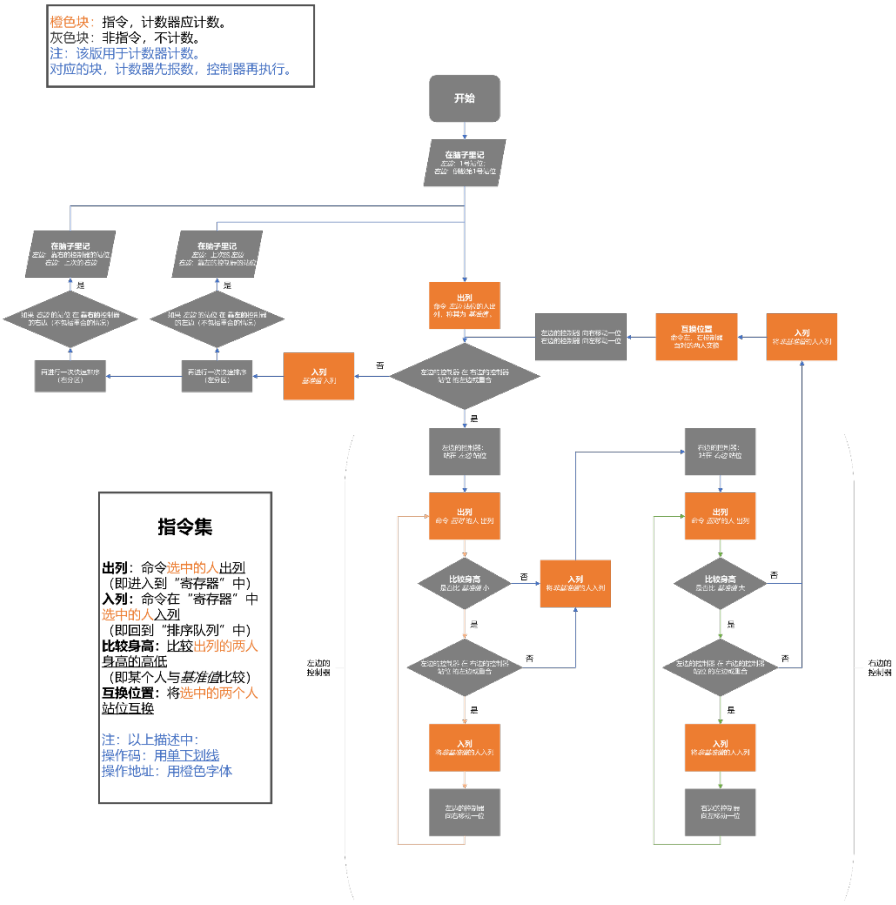
本着人类能理解且易于执行的原则，指令集内仅有三条指令。
（计数器报数）-> 控制器执行 && 监控器拍照 && 监督器检查是否有错
当且仅当执行的是指令时，计数器计数一次，监控器拍摄一次。

① 指令集

出列：命令选中的人出列，即进入到“寄存器”中。
入列：命令在“寄存器”中选中的人入列，即回到“排序队列”中。
互换位置：将选中的两个人的站位互换。
【注】　操作码：用单下划线标出
　　操作地址：用灰色背景标出

② 使用流程图表示

图中用橙色标出的代码块代表指令的执行。



2.2. 自然语言代码

为了不破坏代码块原本格式，在这里直接附上图片。

开始

在脑子里记：*左边*：1号站位；*右边*：倒数第1号站位

执行一次 快速排序

出列：命令 *左边* 站位的人出列，称其为 *基准值* 。

重复进行下述操作，直到发现 左边的控制器 的站位 在 右边的控制器 的站位 的右边。

> 【由左边的控制器执行下列指令】
>
> 站在 *左边* 站位
>
> **出列：**命令 *面对* 的人 出列
>
> **比较身高**
>
> 如果 比 *基准值* 大 且 左边的控制器 的站位 在 右边的控制器 的站位 的左边或重合，则
>
> **入列：**将*非基准值*的人（刚刚挑出来的人）入列
>
> 左边的控制器 向右移动一位
>
> **出列：**命令 *面对* 的人 出列
>
> **比较身高**
>
> 【如果出现比 *基准值* 大或相等 的情况，则继续往下执行】
>
> **入列：**将*非基准值*的人（刚刚挑出来的人 入列）【注：为下一次挑选空位】
>
> 【由右边的控制器执行下列指令】
>
> 站在 *右边* 站位
>
> **出列：**命令 *面对* 的人 出列
>
> **比较身高**

> 如果 比 *基准值* 小 且 左边的控制器 的站位 在 右边的控制器 的站位 的左边或重合，则
>
> **入列：**将*非基准值*的人（刚刚挑出来的人）入列
>
> 右边的控制器 向左移动一位
>
> **出列：**命令 *面对* 的人 出列
>
> **比较身高**
>
> 【如果出现比 *基准值* 小或相等 的情况，则继续往下执行】
>
> **入列：**将*非基准值*的人（刚刚挑出来的人 入列）【注：为互换位置做准备】
>
> 如果 左边的控制器 的站位 在 右边的控制器 的站位 的左边
>
> **互换位置：**命令**左、右控制器**面对的两人交换
>
> 左边的控制器 向右移动一位；右边的控制器 向左移动一位

【注：再分别进行两次 快速排序】

如果 *左边* 的站位 在 **靠左的控制器** 的左边（不包括重合的情况）

在脑子里记：*左边*：上次的*左边*；*右边*：**靠左的控制器**的站位

再执行一次 快速排序

如果 *右边* 的站位 在 **靠右的控制器** 的右边（不包括重合的情况）

在脑子里记：*左边*：**靠右的控制器**的站位；*右边*：上次的*右边*

再执行一次 快速排序

【注：这两个判断确保了下一次进行快速排序的两个分区，一定包含多于两个人。可以想见，如果只有一个人，甚至一个人都没有了，就自然有序，不必排序。】

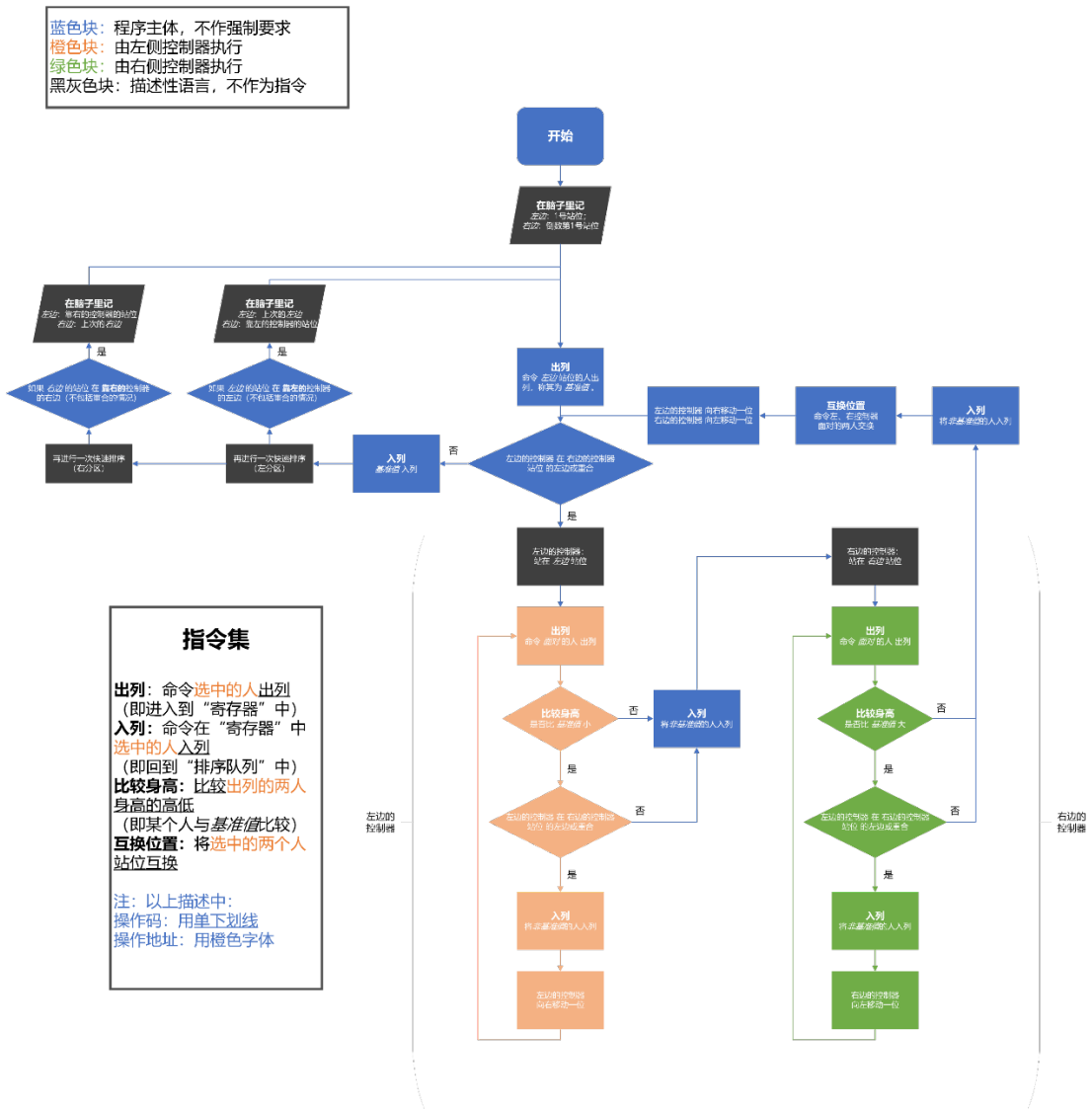
2.3. C 代码原型

使用基于 Hoare 的分区方式，基准值均取每个分区最左侧的数据。

```
void quicksort(int l, int r)
{
    float mid = A[l];
    int i = l, j = r;
    while (i <= j)
    {
        while (A[i] < mid \
        && i <= j) i++;
        while (A[j] > mid \
        && i <= j) j--;
        if (i <= j) {
            temp = A[i];
            A[i] = A[j];
            A[j] = temp;
            i++; j--;
        }
    }
    if (l < j)  quicksort(l, j);
    if (i < r)  quicksort(i, r);
}
```

2.4. 流程图（指令周期）

为便于人类理解，绘制流程图。



2.5. 注意

1. 快速排序方法会产生很多个分区，每个分区又进行一次快速排序。这些分区使用人脑（或纸笔辅助）记忆下来。这个过程中， **在脑子里记** 并不算指令，因为没有对数据操作。每次快速排序只需要记两个位置，命名为 *“左”*、*“右”*。

2. **比较身高** 也不算指令，因为也并未对数据直接操作。借此可比较出列的两人身高的高低（即某个人与*基准值* 比较）

3. 选中的人**“出列”或“入列”**：出列后，在“寄存器”中更便于进行**比较身高**指令。入列指令，保证了“寄存器”中最多只同时存在两人（其中一个人一定是*基准值*）

3. 执行效果与日志

3.1. 实验视频

由于视频文件过大，加之 Word 不便插入视频文件，请从 PPT 中查看

A310-4.mp4（或点击链接在线观看）

3.2. 实验掠影

排序前就位

“凛冬”中的摄影师

左边控制器正在比较身高

右边控制器正在比较身高

监控器正监督结果正确性

排序前，控制组在商议和优化程序运行方案

3.3. 执行效果（日志）

① **运行计时：**整 24 分钟

② **指令计数：**85 + 148 = 233 次

③ **预演次数：**3 + 2 = 5 次

4. 相关思考

4.1. 确保“三个正确性”

① 结果正确

1. **感官上正确**（即在实验进行中当场判断）
   通过控制器的目测比较、确认身高；监督器当场检查，监督控制器比较正确性。
2. **数据本身正确**（即建立一个数据表进行实验后的结果判断）[*]
   事先要求同学们排好身高序号（包含身高相同的情况），检查结果是否相符。

* 注：并非“面向结果的编程”，未改变实验进行中的任何过程，就如对排序算法引入一个已知结果的测试样例，检查其正确性

② 算法正确

1. 由熟练掌握快速排序算法的组员担任*监督器*，严格按照本小组共同商定的算法进行监督，在控制器执行的*每步*判断是否有错。
2. 通过*监控器*录制的*执行记录*进行复盘判断。
3. 换一个人担任*控制器*，代码是可以正常完成的。

③ 系统正确

通过*监督器*在*每步*判断，同时只有一个控制器在运行（非多头领导）
通过*监控器*录制的*执行记录*进行复盘判断。

4.2. 意外情况与寄存器存在的必要性

① 意外情况及处理

前文已提到，将计算机的算法施用于人体身上，思路需要灵活变通，针对问题情境进行转化是很大的难点。

首先，控制器执行算法本身就有难度。虽然我们用自然语言描述了整个程序，但是组员在执行过程中仍然在个别指令执行中有些含糊。为此我们绘制流程图便于理解，并通过研讨、讲解的方式使整个流程清晰。

其次，预演时采用的组员“报身高”方法被认为不可行，我们又重新设定了“寄存器”来直接进行身高比较。具体内容见右侧。

② 寄存器存在的必要性

寄存器位于 CPU 内部，用于存储和传送数据，是暂存 (即非常短暂地读写少量信息并马上用到)。如果没有寄存器，那么计算机的运行效率会大大降低，因为 CPU 需要频繁地访问主存或其他设备，而这些操作的速度远远低于寄存器。

通过将基准值存入寄存器，我们就不必频繁访问数据本身了。(即频繁走到基准值面前，实际上两个距离甚远的人也不好比较身高)

另外，在计算机系统中，寄存器还被用来储存地址和操作数等，是很重要的存储器。

4.3. 代码、动态执行指令与算法的适用性

① 代码和动态执行指令

代码是一种用于表示计算机指令的符号系统，实例有源代码、目标代码、机器代码、字节码、脚本等。

动态执行指令是一种在运行时生成或修改代码的技术，使计算机根据不同的情况或需求，动态地调整或优化行为。动态执行指令的方法有多种，比如解释器、即时编译器、自修改代码、反射、元编程等，它们可以应用于不同的领域。

在人体计算机实验中，我们面向人类能理解的语言编写代码，人类本身有动态执行指令的能力，将代码解释为人的行为。

② 算法的适用性

我们清楚，快速排序算法本身有很好的时间复杂度$O(nlogn)$；是一种原地排序，不需要额外的空间；是不稳定排序，对去重或随机场景有利。通过具体的优化策略(接下来会提到)，可以减免最坏时间复杂度的出现。可以很好胜任数据量很大的情况。

然而，在本组的人体计算机代码中，不该忽略的是如何保存递归信息。计算机可以调用系统栈来保存，人类则需要使用大脑或纸笔记录分区，两者是有差别的。人数较大时，可能带来额外开销，使得对递归分区的记录并不轻松。

4.4. 培养算法思维与系统思维

① 算法思维

算法思维是一种将问题分解为一系列有序、明确和可执行的步骤的思维方式，它强调逻辑性、精确性和效率。算法思维可以让我们设计出高效的程序，实现预期的功能。

通过本次关于“快速排序算法”的实践，我们对算法有了更深层次的认知，而将计算机的算法施用于人体身上，思路需要灵活变通，在这个过程中我们体验到了针对问题情境进行转化的难点。

值得一提的是，由于快速排序算法并不止存在一种，我们通过小组合作，找出了最合适和易于施行的方案，进一步加深了对算法本身的理解。

② 系统思维

系统思维是一种将问题视为一个整体，考虑其各个部分之间的相互作用和影响的思维方式，它强调整体性、关联性和动态性。系统思维可以让我们理解复杂的现象，发现问题的根源和解决方案。

为计算机编程，倾向于站在计算机的角度思考，虽简化了问题，但往往局限了视野，忽视底层的逻辑。

笔者认为，“人体计算机”的相关实验，通过模拟的方式，很好地把计算机系统的各组分映射到了更易于人类可接触的立体空间中。操纵这个复杂系统完成模拟实验，我们对计算机系统的了解变得更加深刻。

5. 相关附件

由于资源较多且不能完美地在 Word 中呈现，提供超链接，您可以点击来在线访问。

① PPT 形式呈现的报告：A310-4 人体快排计算机实验报告.pptx

② 快进处理的实验视频：A310-4.mp4

③ 彩色的流程图（供组员使用）：General-人体计算机快排流程图.pdf

④ 双色的流程图（供计数器使用）：Counter-人体计算机快排流程图.pdf

⑤ 在 C 语言中的实现：QS-JH.c

⑥ 具体快速排序的算法分类：https://blog.jstar.vip/2023/11/07/快速排序算法编程实践

计算机科学与技术导论 A310 - 4 全体成员

倾情奉献

2023 年 11 月 11 日