# 实验二 无符号数组排序

1. 题目要求

在 buf 缓冲区中存放有 50 个字节数据(无符号数)，编写程序将这些数据由小到大排序，排序后的数据仍放在该区域中。具体功能如下:

1. 原始数据在源程序中由定义给出;
2. 在屏幕上先显示排序前的数据（十六进制），数据以每行 10 个的格式显示在屏幕上，数据与数据间用空格分开；
3. 完成排序;
4. 在屏幕上显示排序后的数据（十六进制），数据以每行 10 个的格式显示在屏幕上，数据与数据间用空格分开。

要求:

1. 如果数据个数变为 60 或 78 等的取值时，程序代码部分无需修改;
2. 编写完整程序，并能上机运行;
3. 完成实验报告(格式不限)，写出算法思想、调试过程及心得体会，2-3 页以内。
4. 算法说明

综合考虑时间复杂度、空间复杂度和编码难度，使用快速排序算法。

一轮快速排序：

* 1. 抽取首位数字x作为此轮排序基准，存在寄存器中，将[buf+0]视为空位；设置两个记录遍历到的数组位置的指针i和j，i从首位开始向后移动（初值0），j从末位开始向前移动（初值为数组长度-1）；
  2. j从后往前寻找比x小的数，找到后将符合要求的[buf+j]复制到第i位[buf+i]，则[buf+j]可视为空位，转到状态（3）；
  3. i从前往后寻找比x大的数，找到后将符合要求的[buf+i]放在第j位[buf+j]，则[buf+i]可视为空位，转到状态（2）；
  4. 重复（2）（3）直到i==j，此时[buf+i]为空位，但已经没有需要比较的数字了，且其左侧所有数都小于x，右侧所有数都大于x，将x放入空位，本轮结束。

继续在左侧、右侧两段中进行下一轮快速排序，直到数组长度=1。

1. 流程设计框图
2. 代码块、子程序功能

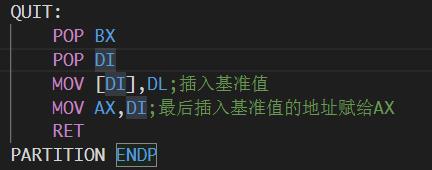
各代码块功能与流程图一一对应。

排序：

1. QSORT：快速排序函数，调用时将左边界DI至右边界BX内（左右均为闭区间）的数组从无序变为有序（从小到大）。
2. PARTITION：进行一轮排序，将左边界DI至右边界BX内（左右均为闭区间）的数据分成两段，左段均小于基准值，右段均大于基准值。其中基准值为输入时的首位。将基准值地址存在AX输出。

输出：

1. PROC SHOW：控制输出格式，每行显示十个两位16进制数字输出并以回车结尾。使用时SI指向要显示的数组。
2. PRINT：输出一个两位16进制数，并在其后添加一个空格，调用时输出内容放在DL中。
3. 调试过程

必须最后pop！

1. 无法模块化测试，难以定位bug
2. 心得体会

函数复用性差，难以做到模块间低耦合