



# 《计算机组成原理实验》 实验报告

(实验一)

学院名称：数据科学与计算机学院

专业（班级）：16 计算机类 4 班

学生姓名：郑映雪

学号：16337327

时间：2017 年 10 月 15 日

成绩：

---

## 实验一：MIPS汇编语言程序设计实验

---

### 一. 实验目的

- (1) 初步认识和掌握 MIPS 汇编语言程序设计的基本方法；
- (2) 熟悉 PCSpim 模拟器的使用。

### 二. 实验内容

从键盘输入 10 个无符号字数或从内存中读取 10 个无符号字数并从大到小进行排序，排序结果在屏幕上显示出来。

### 三. 实验器材

电脑一台，PCSpim 仿真器软件一套。

### 四. 实验过程与结果

①设计思想：冒泡排序的算法思想：

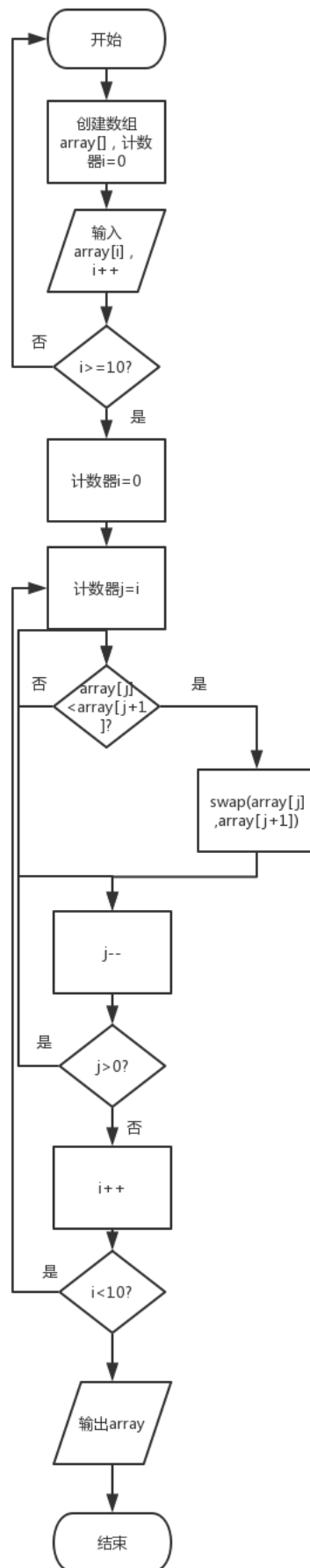
- i. 比较相邻的两个数，得出最小的数放在最后一位。
- ii. 从头比较相邻的两个数，除了最后一个，再把最小的数放在倒数第二

位

- iii. 重复多次直到排序完成

分析得到时间复杂度为  $O(n^2)$

②根据算法画出流程图：



## ③按流程图写出C++代码

```
#include <iostream>

using namespace std;

int main(){

    int a[10],temp;

    for(int i=0;i<10;++i)

        cin >>a[i];

    for (int i=0;i<10;++i){

        for (int j=i;j>0;--j){

            if (a[j]<a[j+1]){

                temp=a[j];

                a[j]=a[j+1];

                a[j+1]=temp;

            }

        }

    }

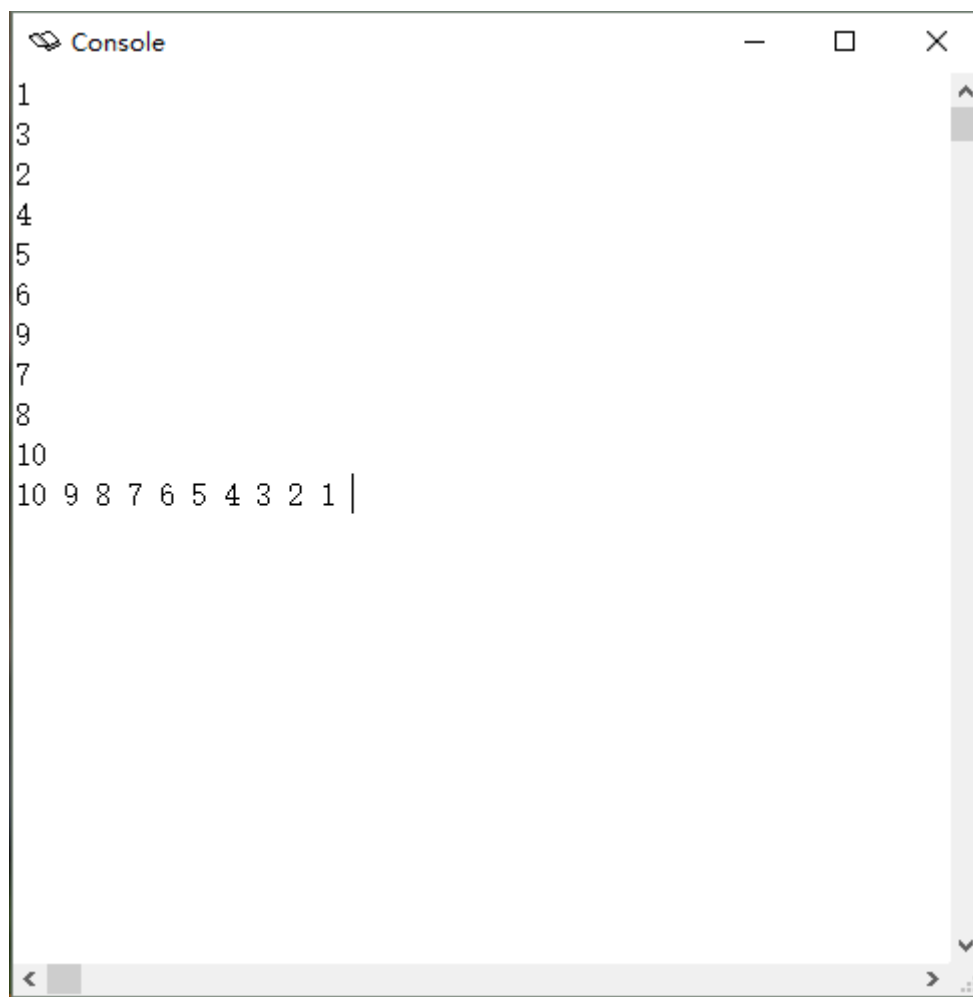
    for (int i=0;i<10;++i)

        cout <<a[i]<<' ';

}
```

## ④根据C++代码写出相应的mips代码（具体代码附于最后）

## ⑤实验结果：



```
Console
1
3
2
4
5
6
9
7
8
10
10 9 8 7 6 5 4 3 2 1 |
```

## 五、实验心得

1、我认识到了mips语言和以前学过的高级语言都不一样，一句高级语言的语句我需要反复思考和选择到底用哪种操作更为简便，所以一个两分钟就能写好的C++排序代码，我用mips语言写了两个小时。在mips代码的编写中，我得对内存结构思考清楚再编写，它不像高级语言一样智能，便更要求我们对内存的相关知识更加熟悉。

2、一开始写的时候，10个数排序总会莫名其妙出现一个32，原本的十个数会缺少一个数，于是先在C++语言里改，发现怎么也找不出算法的漏洞，后来我

想到了数组分配空间的问题，多分配了一个空间防止溢出，问题就解决了。我又回到mips里，将代码改成.space 44（即比原来多开一个元素空间），问题果然解决了。学高级语言时老师就提醒我们要考虑防止溢出，在汇编语言里我也得一样重视这个问题。

3、以前写冒泡排序时变量j我习惯从0到n-i-1递增，但我发现这样在mips里会麻烦很多，我又发现如果j从i开始计数再递减到0是一样的效果，而这样又在mips里简便很多，所以我就采用了j递减的方案，后期检查的时候也方便了许多。

### 【程序代码】

```
.text
.globl main

main:
    la $t0,array
    add $t1,$zero,$t0
    addi $t7,$t0,40    #设置数组起始地址和终止地址

    addi $t2,$zero,0    #设置计数器
input:
    li $v0,5
    syscall    #输入10个数
    sw $v0,0($t1)
```

```
    addi $t1,$t1,4      #计数器依次遍历
    addi $t2,$t2,1
    slti $s0,$t2,10
    bnez $s0,input      #若不满10次则继续输入

    addi $t3,$zero,0    #外层循环的计数器i
loop1: #外循环
    add $t1,$zero,$t0   #每次内循环比较完一轮后，指针指向array[0]的地址
    slti $s0,$t3,10
    beqz $s0,print      #循环满10次则跳出处理

    addi $t4,$t3,0      #内层循环的计数器j
loop2:
    slti $s0,$t4,0
    bnez $s0,exitloop   #判断是否继续进行内层循环

    sll $t6,$t4,2       #左移j两位使之*4
    add $t6,$t6,$t1      #t6是数组的第j位元素的地址，预先存储方便交换
    lw $t8,0($t6)        #t8=array[j]
    lw $t9,4($t6)        #t9=array[j+1]
    slt $s0,$t8,$t9
    bnez $s0,swap        #比较，若小于则交换
    addi $t4,$t4,-1      #j--
    j loop2

swap:
    sw $t8,4($t6)        #利用预先的$t6对两个寄存器进行交换
    sw,$t9,0($t6)
    addi $t4,$t4,-1
```

```
j loop2

exitloop:
    addi $t3,$t3,1    #i++
    j loop1

print:
    lw $a0,0($t0)    #打印数据
    li $v0,1
    syscall

    la $a0,space      #打印空格
    li $v0,4
    syscall

    addi $t0,$t0,4
    bne $t0,$t7,print

j exit
exit:
    li $v0,10
    syscall

.data
array:
    .space 44    #为数组分配空间，多分配一位防止溢出
space:
    .ascii " "    #数据输出时用空格隔开
```