实验二: 分支程序实验

实验环境

PC 机 + Win 2003 + emu8086

仅供参考 严禁抄袭 Bolg:zhangshier.vip

一. 实验项目要求

熟悉 emu8086仿真系统

实现连续地址的内存数据访问

用两种方法实现对五个字从大到小排序

扩展要求:

用冒泡法实现快排

计算不同方法的复杂度

二. 理论分析或算法分析(含实验项目要求的分析、数学或逻辑推导等)

选择排序:第一次从 $R[0]^R[n-1]$ 中选取最大值,与 R[0] 交换,第二次从 $R[1]^R[n-1]$ 中选取最大值,与 R[1] 交换,……,第 i 次从 $R[i-1]^R[n-1]$ 中选取最大值,与 R[i-1] 交换,……,第 n-1 次从 $R[n-2]^R[n-1]$ 中选取最大值,与 R[n-2] 交换,总共通过 n-1 次,得到一个按排序码从大到小排列的有序序列。

排序前: 93642

第一次: 93642

第二次: 96342

第三次: 96432

第四次: 96432

冒泡排序: 重复地遍历要排序的数列,第一次比较 $R[0]^R[n-1]$ 中相邻两个元素是否满足 R[i]>R[i+1] 如果不满足就交换,第二次比较 $R[0]^R[n-2]$,....,第 n-1 次比较 $R[0]^R[1]$,一 共比较 n-1 次

排序前: 23649

第一次: 36492

第二次: 64932

第三次: 69432

第四次: 96432

扩展要求: 冒泡法实现快排,即在冒泡排序的基础上,设置一个标志位,如果当前内循环没有做交换,说明排序正确,直接跳出循环

排序前: 93642

第一次: 96432

第二次: 96432 没有做交换,直接跳出循环结束程序

三. 实现方法(含实现思路、程序流程图、实验电路图和源程序列表等)

选择排序

MOV CX, 5-1 ;选择排序,外循环 N-1 次

MOV BX, OFFSET TAB

LP1:

MOV AX, [BX] ;取当前 TAB[i]给 AX PUSH CX ;保存数据,CX压栈

MOV SI, BX :暂存BX给SI

LP2:

ADD SI, 2 ;偏移地址+2 转向下一个数

CMP AX, [SI] ;相邻两个数比较 ;如果 AX>[SI]跳转 JAE J1

;如果AX<[SI]交换,最后一次内循环找到最大数AX XCHG AX, [SI]

J1:

LOOP LP2 :继续回到 LP2 内循环 MOV [BX], AX ;找到的最大数给[BX]

POP CX ADD BX, 2

LOOP LP1 ;回到外层循环

HLT

TAB DW 9, 3, 6, 4, 2 ; 段结构使用数据变量放在后面, 定义五个数

冒泡排序

MOV CX, 5-1 ; 冒泡排序, 外循环 N-1 次

LP1: PUSH CX

MOV SI, OFFSET TAB ;当前 TAB 指针偏移地址给 SI

LP2:

MOV AX, [SI]

CMP AX, [SI+2] ;比较 TAB[i]和 TAB[i+1] JAE J1 ;TAB[i]>TAB[i+1]跳转 XCHG AX, [SI+2] ;TAB[i]<TAB[i+1]交换

MOV [SI], AX

J1:

;SI+2 指向下一个数据 ADD SI, 2 LOOP LP2 ;继续下一次内循环

2

esono,

POP CX

LOOP LP1 ;内循环完成,进行外循换

;JMP \$

HLT

TAB DW 9, 3, 6, 4, 2 ;段结构使用数据变量放在后面,定义五个数

冒泡法快速排序

MOV CX, 5-1 ; 冒泡法快速排序, 外循环 N-1 次

LP1:

PUSH CX

MOV SI, OFFSET TAB ; 当前 TAB 指针偏移地址给 SI

MOV BX, 0 ;设置标志位如果做交换就+1,如果本次循环没有做交换(BX==0)说明排

序正确, 跳出内外循环结束程序

LP2:

MOV AX, [SI]

 CMP AX, [SI+2]
 ;比较 TAB[i]和 TAB[i+1]

 JAE J1
 ;TAB[i]>TAB[i+1] 跳转

 XCHG AX, [SI+2]
 ;TAB[i]<TAB[i+1]交换</td>

MOV [SI], AX

ADD BX, 1

J1:

ADD SI, 2 ;SI+2 指向下一个数据 LOOP LP2 ;继续下一次内循环

POP CX CMP BX, 0

JNA J2 ;BX 不大于 0 跳转

LOOP LP1 ;内循环完成,进行外循换

;JMP \$

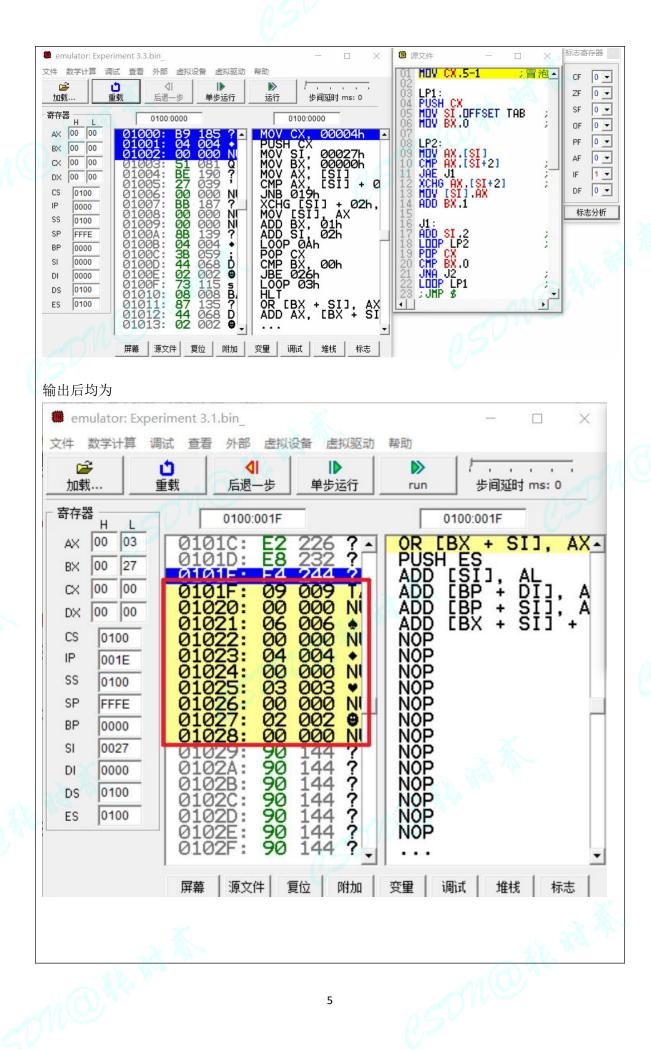
J2:

HLT

TAB DW 9, 3, 6, 4, 2 ;段结构使用数据变量放在后面,定义五个数

3

四. 实验结果分析(含执行结果验证、输出显示信息、图形、调试过程中所遇的问题及处理方法等,如果 有引用的参考文献,安排在本节最后列出) 选择排序 emulator: Experiment 3.1.bin_ ■ 源文件 文件 数学计算 调试 查看 外部 虚拟设备 虚拟驱动 帮助 MOV CX.5-1 MOV BX.OFFSET TAB 4 0 🕶 步间延时 ms: 0 LP1: MOV AX.[BX] PUSH CX MOV SI.BX 0 -寄存器 0100:0000 0 🔻 MOV CX, 00004h
MOV BX, 0001Fh
MOV AX, [BX]
PUSH CX
MOV SI, 02h
CMP AX, [SI]
JNB 014h
XCHG [SI], AX
LOOP 08h
MOV [BX], AX
POP CX
ADD BX, 02h
LOOP 06h
HLT
OR [BX + SI], AX
ADD AX, [BX + SI
PUSH ES
ADD [SI], AL OF 00 00 ΔX : 暂存 0 🕶 00 00 0 🕶 00 00 CX ;偏移;相邻果 1 🔻 IF DX 00 00 01005: 01006: 01008: 01009: 01008: 01008: 01008: 01000: 01000: 01000: 01001: 01011: 01011: 01013: Z?BQ????• 0 -0100 XCHG AX.[SI] ;如果 IΡ 0000 标志分析 SS 0100 :继续:找到 FFFE BP 0000 SI 0000 ;回到 0000 DI HLT 0100 0100 屏幕 | 源文件 | 复位 | 附加 | 变量 调试 堆栈 标志 冒泡排序, LOP1 执行了 n-1 次 memulator: Experiment 3.2.bin ■ 源文件 文件 数学计算 调试 查看 外部 虚拟设备 虚拟驱动 帮助 HOV CX.5-1 0 -ZF 步间延时 ms: 0 加载. CX SI.OFFSET TAB SF 0 🕶 寄存器 0100:0000 0100:0000 MOV CX, 000004
PUSH CX
MOV SI, 000104
MOV AX, [SI]
CMP AX, [SI]
JNB 013h
XCHG [SI] + 0
MOV [SI], AX
ADD SI, 02h
LOOP 07h
POP CX
LOOP 03h
HLT 0 🕶 OF AX 00 00 0 🕶 00 00 ВX 0 -00 00 $\mathbb{C}\!\!\times\!$ 081 190 028 009 0059 0059 0055 0035 0035 1068 1304 131 1 🕶 DX 00 00 01005: 01006: 01008: 01009: 01008: 01008: 01008: 01000: 01000: 01000: 01001: 01011: 01011: 01013: Z? + .D9 54?D9? +? 0 -CS 0100 02h. IP 0000 标志分析 SS 0100 FFFE BP 0000 SI 0000 DI 0000 HLT 0100 TAB DW 9.3.6.4.2 0100 屏幕 源文件 复位 附加 变量 调试 堆栈 250110 30 A 冒泡法快速排序,LP1 在执行第二次循环结束后,跳出程序 USONON 50110 M. Ha



CSONO, AV

复杂度

时间复杂度

选择排序

 $O(N^2)$

冒泡排序

 $O(N^2)$

冒泡法快速排序 O(N)

思考内容

USONO NO NO NO

冒泡法属于快速排序的原因:如果相邻两个数的大小满足条件不需要排序,可以直接跳出程序。

esono M. R

6

CSDN ON