# 《汇编语言》作业七 实践课程中的程序

**班级：计科173班 学号：1707310248 姓名：孟维民**

## 题目

实现冒泡排序、直接插入排序、选择排序。

## 实验准备

2.1实验数据

datasg segment;字节型数据16个，占16个字节

array db -4, 0, 27, 102, -10, -28, 47, 36, 0, -10, -12, 35, 7, -4, -15, 8

len equ($ - array);记录数据个数

datasg ends

Stacksg segment;栈区7个字，占16个字节

dw 0, 0, 0, 0, 0, 0, 0, 0

Stacksg ends

swap marco idata;交换两个数宏

mov al,[bx]

mov ah,[bx + idata]

mov [bx],ah

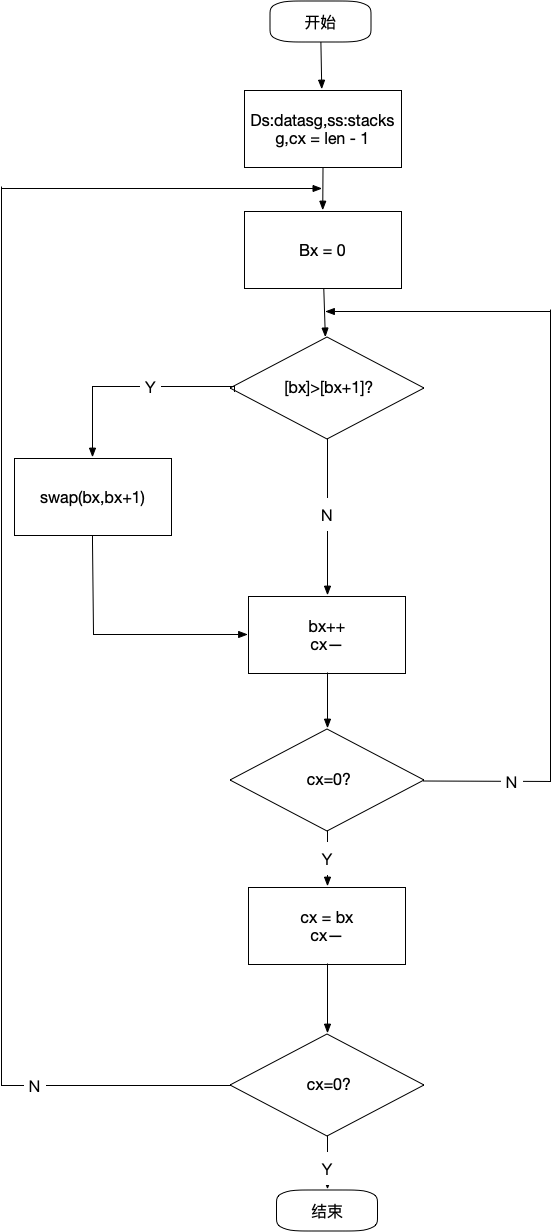
mov [bx + idata],al

2.2冒泡排序

冒泡排序的思路是，外循环执行len - 1次，第i遍外循环时内循环把第bx个元素和第bx + 1（idata = 1）个元素比较，前面的元素较大则交换，一共比较len - 1次。

ds存储数据段，ss存储栈，bx从偏移量0开始，比较[bx]和[bx + 1]，每次增加一，cx 开始等于len - 1，每次内循环完成后把bx - 1的值赋给cx表示外层循环执行次数。

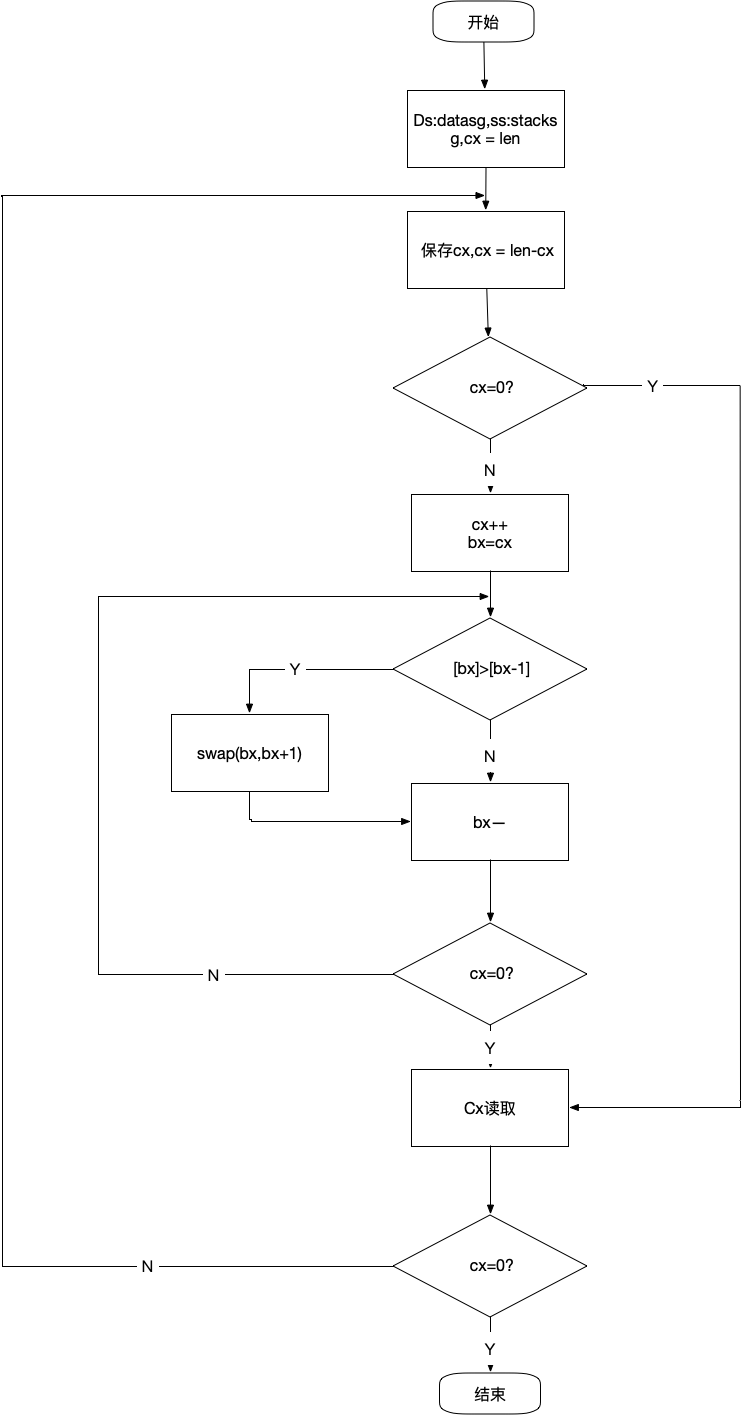
流程图如下：

图2-1 冒泡排序流程图

2.3直接插入排序

直接插入排序的思路是，外循环执行len次，第i遍外循环时先使ax=len-cx，再将cx存入栈中，cx = ax，如果cx = 0，本轮不比较跳转，bx = cx，内循环把第bx个元素和bx - 1（idata = -1）个元素比较，前面的元素较大则交换，一共比较cx次。回到外循环前出栈cx。

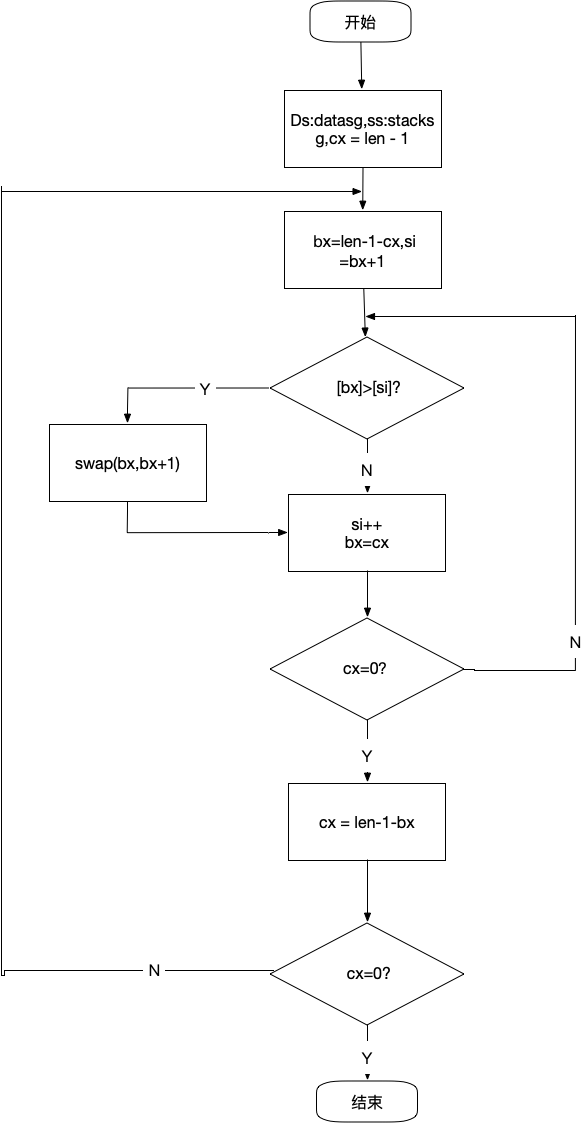
流程图如下：

图2-2 直接插入排序流程图

2.4选择排序

直接插入排序的思路是，外循环执行len - 1次，第i遍外循环时先使bx = len - 1 - cx，si = bx，内循环把第bx个元素和第si（idata = si-bx）个元素比较，前面的元素较大则交换，一共比较cx次。回到外循环前cx = len - 1 - bx。

流程图如下：

图2-3 选择排序流程图

## 实验结果

3.1冒泡排序

3.1.1源代码

datasg segment

array db -4,0,27,102,-10,-28,47,36,0,-10,-12,35,7,-4,-15,8

len equ($-array)

datasg ends

stacksg segment

dw 0,0,0,0,0,0,0,0

stacksg ends

swap macro idata

mov al,[bx]

mov ah,[bx+idata]

mov [bx],ah

mov [bx+idata],al

endm

assume cs:codesg

codesg segment

start: mov ax,datasg

mov ds,ax

mov ax,stacksg

mov ss,ax

mov sp,10h

mov ax,len-1

mov cx,ax

s1: mov bx,offset array

s0: mov al,[bx+1]

cmp byte ptr [bx],al

jg s

conti: inc bx

loop s0

mov cx,bx

loop s1

mov ax,4c00h

int 21h

S: swap 1

jmp conti

codesg ends

end start

3.1.2 运行结果



图3-1 执行前数据

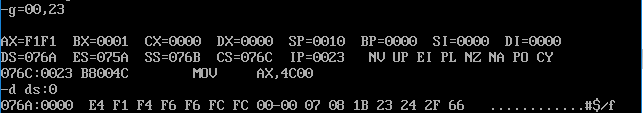


图3-2 执行后数据

3.2直接插入排序

3.2.1源代码

datasg segment

array db -4,0,27,102,-10,-28,47,36,0,-10,-12,35,7,-4,-15,8

len equ($-array)

datasg ends

stacksg segment

dw 0,0,0,0,0,0,0,0

stacksg ends

swap macro idata

mov al,[bx]

mov ah,[bx+idata]

mov [bx],ah

mov [bx+idata],al

endm

assume cs:codesg

codesg segment

start: mov ax,datasg

mov ds,ax

mov ax,stacksg

mov ss,ax

mov sp,10h

mov cx,len

s1: mov ax,len

sub ax,cx

push cx

mov cx,ax

loopz conti2

inc cx

mov bx,cx

s0: mov al,[bx-1]

cmp byte ptr [bx],al

jl s

conti1: dec bx

loop s0

conti2: pop cx

loop s1

mov ax,4c00h

int 21h

S: swap -1

jmp conti1

codesg ends

end start

3.2.2 运行结果



图3-3 执行前数据

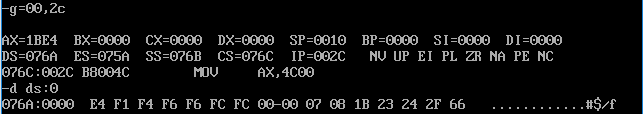


图3-4 执行后数据

3.3选择排序

3.3.1源代码

datasg segment

array db -4,0,27,102,-10,-28,47,36,0,-10,-12,35,7,-4,-15,8

len equ($-array)

datasg ends

stacksg segment

dw 0,0,0,0,0,0,0,0

stacksg ends

swap macro idata

mov al,[bx]

mov ah,[bx+idata]

mov [bx],ah

mov [bx+idata],al

endm

assume cs:codesg

codesg segment

start: mov ax,datasg

mov ds,ax

mov ax,stacksg

mov ss,ax

mov sp,10h

mov cx,len-1

s1: mov ax,len-1

sub ax,cx

mov bx,ax

inc ax

mov si,ax

s0: mov al,[si]

cmp byte ptr [bx],al

jg s

conti: inc si

loop s0

mov ax,len-1

sub ax,bx

mov cx,ax

loop s1

mov ax,4c00h

int 21h

S: mov di,si

sub di,bx

swap di

jmp conti

codesg ends

end start

3.3.2 运行结果



图3-5 执行前数据

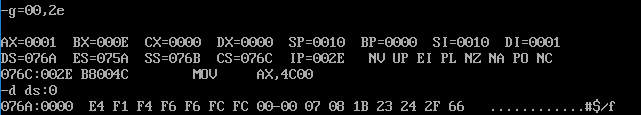


图3-6 执行后数据