

Lab3 实验报告

PB19151769 马宇骁

算法部分：

在 C 语言程序设计 A 和数据结构中我们接触过排序算法，根据 LC3 特性，发现采用快速排序，归并排序或者堆排序等时间复杂度低的排序算法代码长度均过长，不符合 ICS 中行数尽可能少的特点。因此，综合利弊比较，采用最基础的冒泡排序能减少行数。想法：依次读取地址所储存的值比较，进行交换，经过循环，直到最后结点为空，小循环结束，小循环一系列结束一次最大值放末尾，大循环执行第一次，大循环执行完排序完成。由此思路编写算法如下。

编写部分：

根据书中学过的汇编语言，依据算法写出如下代码：

```
.ORIG x3000
LD    R0,BEGIN          ; 第一个点为空，结束
BRz   END
LDR   R5,R0,#0          ; 第二个节点为空，结束
BRz   END

FLAG2  LDR   R2,R0,#0      ; R2 为下一个地址

FLAG1  LDR   R1,R0,#1      ; R1 是当前节点的值
      LDR   R3,R2,#1      ; R3 是下一个节点的值

COMPARE NOT    R4,R1      ; 比较 R1 和 R3
      ADD   R4,R4,#1
      ADD   R4,R4,R3
      BRzp  JUMP

      STR   R3,R0,#1      ; R3<R1,则交换两个节点的值(地址不变)
      STR   R1,R2,#1

JUMP   LDR   R2,R2,#0      ; R2(下一个地址)为空
      LDR   R5,R2,#0
      BRz   JUDGE
      BRnzp FLAG1

JUDGE  LDR   R0,R0,#0      ; R0 后移，R0 移动到尾时，结束
      LDR   R5,R0,#0
      LDR   R5,R5,#0
      BRz   END
      BRnzp FLAG2        ; R0 没到尾，则继续大循环
```

```

END      HALT
BEGIN    .FILL X3100
        .END

```

测试部分：

!	▶	x3100	x3102	12546
!	▶	x3101	x4444	17476
!	▶	x3102	x3106	12550
!	▶	x3103	x2222	8738
!	▶	x3104	x0000	0
!	▶	x3105	x0000	0
!	▶	x3106	x310A	12554
!	▶	x3107	x1111	4369
!	▶	x3108	x0000	0
!	▶	x3109	x0000	0
!	▶	x310A	x0000	0
!	▶	x310B	x0000	0
!	▶	x310C	x0000	0

R0	x3100	12544	
R1	x7FFF	32767	
R2	x0000	0	
R3	x0000	0	
R4	x0000	0	
R5	x0000	0	
R6	x0000	0	
R7	x0000	0	
PSR	x8001	-32767	CC: P
PC	x3001	12289	
MCR	x0000	0	

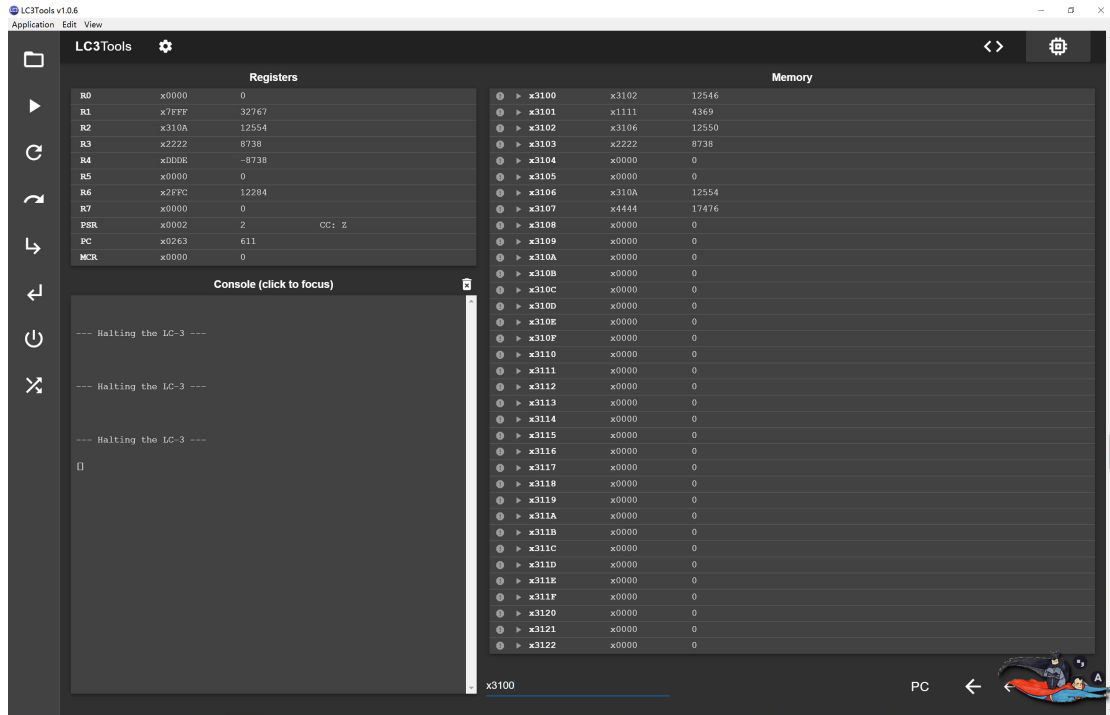
R0	x3100	12544	
R1	x7FFF	32767	
R2	x0000	0	
R3	x0000	0	
R4	x0000	0	
R5	x3102	12546	
R6	x0000	0	
R7	x0000	0	
PSR	x8001	-32767	CC: P
PC	x3003	12291	
MCR	x0000	0	

R0	x3100	12544	
R1	x4444	17476	
R2	x3102	12546	
R3	x0000	0	
R4	x0000	0	
R5	x3102	12546	
R6	x0000	0	
R7	x0000	0	
PSR	x8001	-32767	CC: P
PC	x3006	12294	
MCR	x0000	0	

R0	x3100	12544	
R1	x4444	17476	
R2	x3102	12546	
R3	x2222	8738	
R4	xBBBC	-17476	
R5	x3102	12546	
R6	x0000	0	
R7	x0000	0	
PSR	x8004	-32764	CC: N
PC	x3009	12297	
MCR	x0000	0	

R0	x3100	12544	
R1	x2222	8738	
R2	x3106	12550	
R3	x1111	4369	
R4	xDDDE	-8738	
R5	x310A	12554	
R6	x0000	0	
R7	x0000	0	
PSR	x8001	-32767	CC: P
PC	x3007	12295	
MCR	x0000	0	

经过一步一步顺序执行，结果如下：



在第一次代码试运行均正确输出，由此算法代码正确！