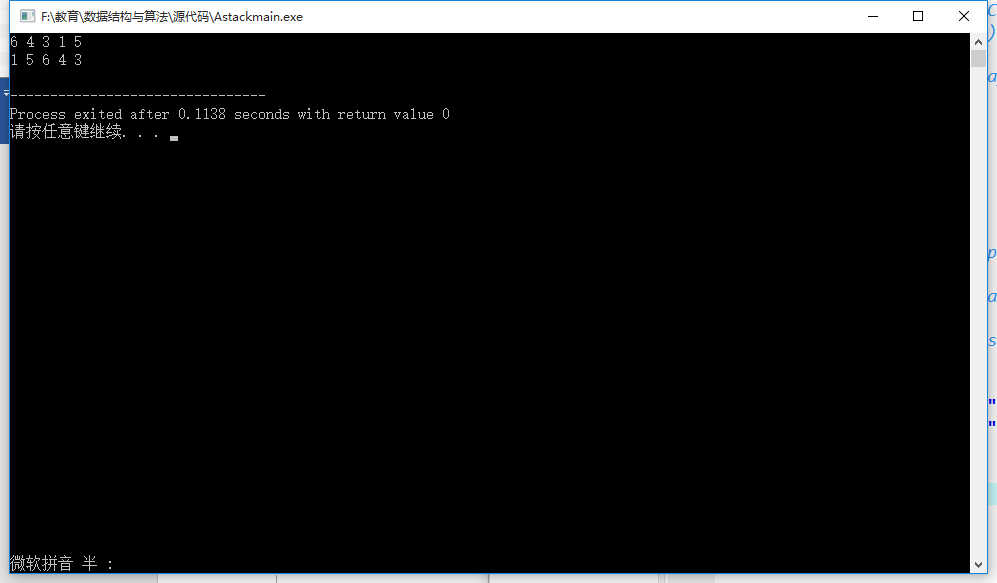
20161607\_沈志立\_2班\_第六周作业

7.2

开发环境:DEV-C++

1. 将下段代码放进astack.h
2. **void** StackSort(AStack<**int**>& Input){
3. AStack<**int**> TempA,TempB;
5. **while**(!Input.isEmpty())
6. TempA.push(Input.pop()) ;
7. Input.push(TempA.pop());
8. **while**(!TempA.isEmpty())
9. **while**(Input.top()>TempA.top())
10. TempB.push(Input.pop());
11. Input.push(TempA.pop());
12. **while**(!TempB.isEmpty())
13. Input.push(TempB.pop());
14. }
16. **bool** isEmpty(){
17. **if**(top =0) **return** **true**;
18. }
19. 将下端代码放进main.cpp
20. #include "astack.h"
22. **int** main() {
24. AStack<**int**> S1;
26. S1.push(5);
27. S1.push(1);
28. S1.push(3);
29. S1.push(4);
30. S1.push(6);
32. **while** (S1.length() > 0) {
33. cout << S1.pop() << endl<<"\n";
34. }
36. S1.StackSort(S1);
38. **while** (S1.length() > 0) {
39. cout << S1.pop() << endl;
40. }

43. **return** 0;
44. }
45. 运行结果



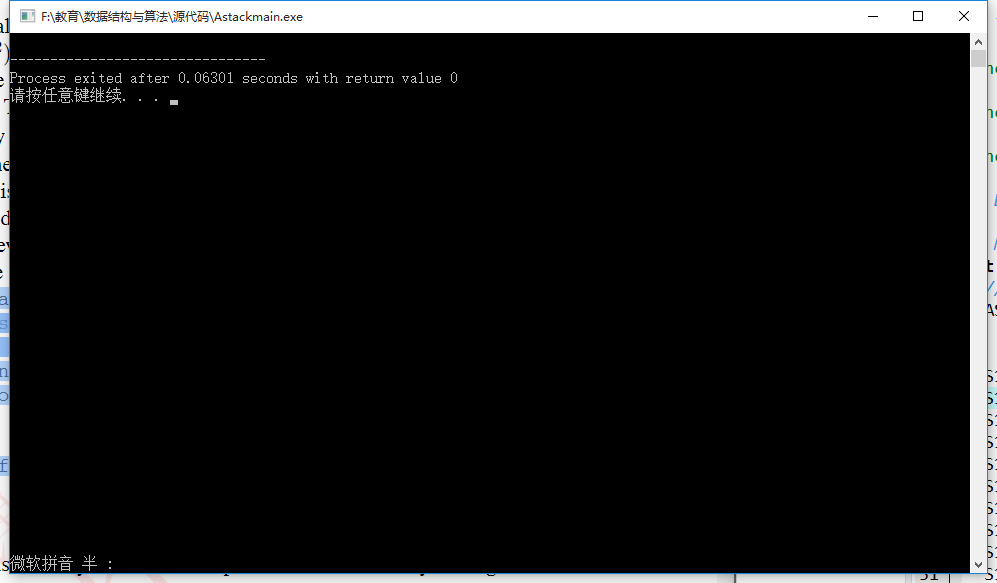
7.5

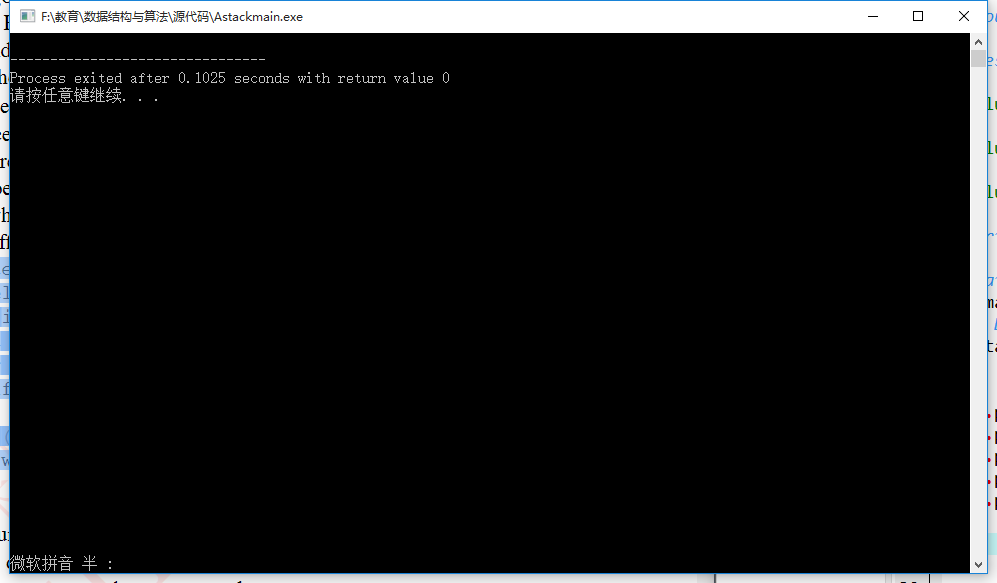
(a)

1. **template** <**class** Elem, **class** Comp>
2. **void** selsort(Elem A[], **int** n) {
3. **for** (**int** i=0; i<n-1; i++) {
4. **int** lowindex = i;
5. **for** (**int** j=n-1; j>i; j--)
6. **if** (Comp::lt(A[j], A[lowindex])) lowindex = j;
7. **if** (i != lowindex)
8. swap(A, i, lowindex);
9. }
10. }

(b)这种改进会减缓算法的速度 因为它的检查也是需要时间的

(c)原来的运行速度



改进后的运行速度

7.6

插入排序是稳定的 因为在较低位置元素比其小的情况下才交换

冒泡排序是稳定的 因为在较低位置元素比其小的情况下才交换

选择排序不是稳定的 相等的情况下也会发生交换

Shell排序不是稳定的 相等的情况下也会发生交换

快速排序不是稳定的 在中心点选定之后，它会交换最后一个元素，相等的情况下也会发生交换