黑盒测试：

不充分，还需要以下测试用例

1.异常情况测试用例：

输入A=0,B=0,C=0，预期输出为无数解。

输入A=0,B=0,C='c'，预期输出为输入有误。

输入A=1,B=2,C=3，预期输出为无实数解。

输入A=1,B=2,C=3，方程不再是一元二次方程，预期输出一个解。

2.边界测试用例：

输入A=1,B=1,C=1e-15，预期输出为无解。

输入A=1,B=1,C=1e-16，预期输出为无解。

输入A=1,B=1,C=1e-17，预期输出为无解。

输入A=1,B=1,C=-1e-15，预期输出为-1和-1。

输入A=1,B=1,C=-1e-16，预期输出为-1和-1。

输入A=1,B=1,C=-1e-17，预期输出为-1和-1。

白盒测试：

所有路径测试有4条：

1-2-3-4-5-6-7

1-2-4-5-6-7

1-2-3-4-5-6-1

1-2-4-5-6-1

虽然所有路径都测试通过了，但不能保证程序不会出错。因为白盒测试只是对程序内部的代码进行测试，可能存在以下情况：

1.测试用例未覆盖所有可能的输入，导致某些情况未被测试到。

2.程序存在逻辑错误或算法错误，导致即使所有路径都被测试通过，仍然存在某些输入会导致程序出错。

3.程序存在未处理的异常情况，例如除以零、越界等，可能会导致程序崩溃或产生错误结果。

因此，虽然所有路径都测试通过了，但还需要进行更全面的测试，包括边界测试、异常情况测试、性能测试等，才能更好地保证程序的正确性。

习题7

自底向上方法：

1)Test E,G,H,J,K,L,M,N

2)Test (F,L),(I,M,N)

3)Test (B,F,L,G),(C,H),(D,I,J,K,M,N)

4)Test (A,B,C,D,E,F,G,H,I,J,K,L,M,N)

自顶向下方法：

1)Test A

2)Test (A,B,C,D,E)

3)Test (A,B,C,D,E,F,G,H,I,J,K)

4)Test (A,B,C,D,E,F,G,H,I,J,K,L,M,N)

改进的自顶向下方法：

1)Test A

2)Test B,C,D,E

3)Test (A,B,C,D,E)

4)Test F,G,H,I,J,K

5)Test (A,B,C,D,E,F,G,H,I,J,K)

6)Test L,M,N

7)Test (A,B,C,D,E,F,G,H,I,J,K,L,M,N)

一次性集成方法：

1)Test A,B,C,D,E,F,G,H,I,J,K,L,M,N

2)Test (A,B,C,D,E,F,G,H,I,J,K,L,M,N)

三明治方法：

1)Test A,L,M,N

2)Test (A,B),(A,C),(A,D),(A,E),(F,L),(I,M,N)

3)Test (B,F,G),(C,H),(D,I,J,K)

4)Test (A,B,C,D,E,F,G,H,I,J,K,L,M,N)

改进的三明治方法：

1)Test A,B,C,D,E,F,I,L,M,N

2)Test G,H,J,K,(A,B),(A,C),(A,D),(A,E),(F,L),(I,M,N)

3)Test (B,F,G),(C,H),(D,I,J,K)

4)Test (A,B,C,D,E,F,G,H,I,J,K,L,M,N)