Basic Coding, Unit Test & Refactoring

课堂内容

- 熟悉解决问题的流程
  1. 分析问题
  2. 形成方案
  3. 探索、尝试解决问题
  4. 确保质量 （单元测试）
  5. 重构程序以满足不断变化的需求
  6. （回到第一步，解决新问题）

练习1：数组中最大的子数组之和

- 输入，一个数组，和它的大小
- 输出，这个数组中最大子数组的和

- 例如

输入	输出
[-1, 2, 3, -4]	5
[-1, 2, -5, 3, -4]	3
[-1, 20, -5, 30, -4]	45
[-2, -3, -5, -1, -9]	-1

算法的效率

- 下一步
  - 从文件中读输入的数据，熟悉文件操作
  - 文件有两种数据
    - 这次测试中有多少个数据； 每个数据的值，用逗号隔开
- 更复杂的例子
  - Input
    -32, -10, 33, -23, 32, -12, 41, -12, 1, 3, 5, -98, 70, -21, 10, -9, 61
  - Output
    sum = 71
- 你的算法效率如何?
  - O(N^3)
  - O(N^2)
  - O(N)

用类/函数来实现

如果设计了一个类 Class

MSA (Maximum Sub-array Sum)

里面有一个函数

Calc()

如何设计这个函数

输入参数是什么，输出是什么

请马上动手做题

Examples of unittest

```
int[] test1 = { 1, -2, 3, 5, -3, 6, 1, -1 };
   msa.Calc(test1, 8);
   Debug.Assert(msa.MaxSumPosition.sum == 12);
   Debug.Assert(msa.MaxSumPosition.start == 2);
   Debug.Assert(msa.MaxSumPosition.end == 6);
```

单元测试练习

- Implement a C# class, or C++ class
- Describe unit test cases
- Performance of your implementation
  - Speed
  - Space
- Submit to Teaching Assistant
- 参见 《构建之法》第二章单元测试的内容

同学们用20 分钟写一个单元测试

好的单元测试

- 检查单元测试是否满足下面的条件
  - 单元测试应该在最基本的功能/参数上验证程序的正确性。
  - 单元测试过后，机器状态保持不变。
  - 单元测试应该产生可重复、一致的结果。
  - 单元测试应该覆盖所有代码路径。

回归测试

- 回归测试 （Regression Test）
- Regress 的英语定义是： return to a worse or less developed state，是倒退、退化、退步的意思。
- 在软件项目中，如果一个模块或功能以前是正常工作的，但是在一个新的构建中出了问题，那么这个模块就出现了一个“退步”（Regression），从正常工作的稳定状态退化到不正常工作的不稳定状态。

扩展 - 锻炼各种能力

- 简单的扩展
  - 如果元素的数值非常大，需要注意什么?
  - 如果一维数组很长，需要注意什么?
- 功能的扩展
  - 把数据放在文件里面，从文件中读数据
  - 把这个程序放到网上去
- 数据量的扩展
  - 如果元素的个数超过10,000 个? 超过1百万个?
- 维度的扩展
  - 二维，三维，首尾相连
  - 考虑程序如何展现这些“子数组”

简单的测试要点

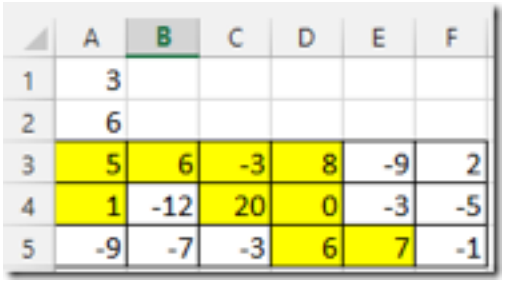
要写多少测试用例才够呢?

- Right-BICEP方法:
  - **R**ight－结果是否正确?
  - **B**order Condition－是否所有的边界条件都是正确的?
  - **I**nverse Relation: 能查一下反向关联吗?
  - **C**ross check: 能用其他手段交叉检查一下结果吗?
  - **E**rror－你是否可以强制错误条件发生?
  - **P**erformance－是否满足性能要求?
- 看大部分代码是否被覆盖了

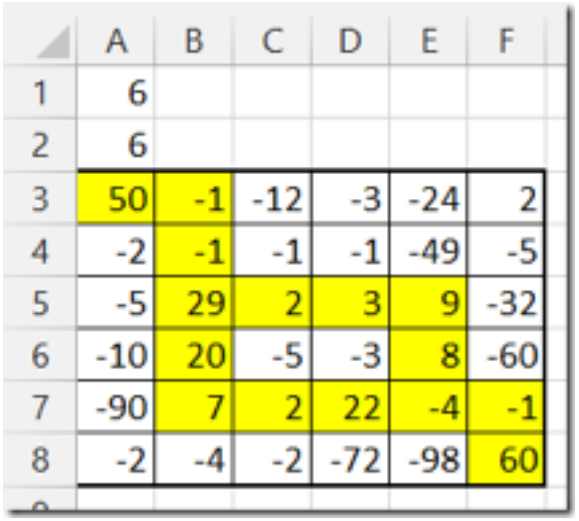
http://www.manongjc.com/article/78833.html

扩展 – 算法的扩展

- 二维数组中【子数组】定义的扩展
  - 传统: 矩形的子数组
  - 扩展: 凡是相连就认为是子数组
- 参考:
  - http://www.cnblogs.com/xinz/p/3318230.html



	A	B	C	D	E	F
1	3					
2	6					
3	5	6	-3	8	-9	2
4	1	-12	20	0	-3	-5
5	-9	-7	-3	6	7	-1



	A	B	C	D	E	F
1	6					
2	6					
3	50	-1	-12	-3	-24	2
4	-2	-1	-1	-1	-49	-5
5	-5	29	2	3	9	-32
6	-10	20	-5	-3	8	-60
7	-90	7	2	22	-4	-1
8	-2	-4	-2	-72	-98	60

扩展 2D array

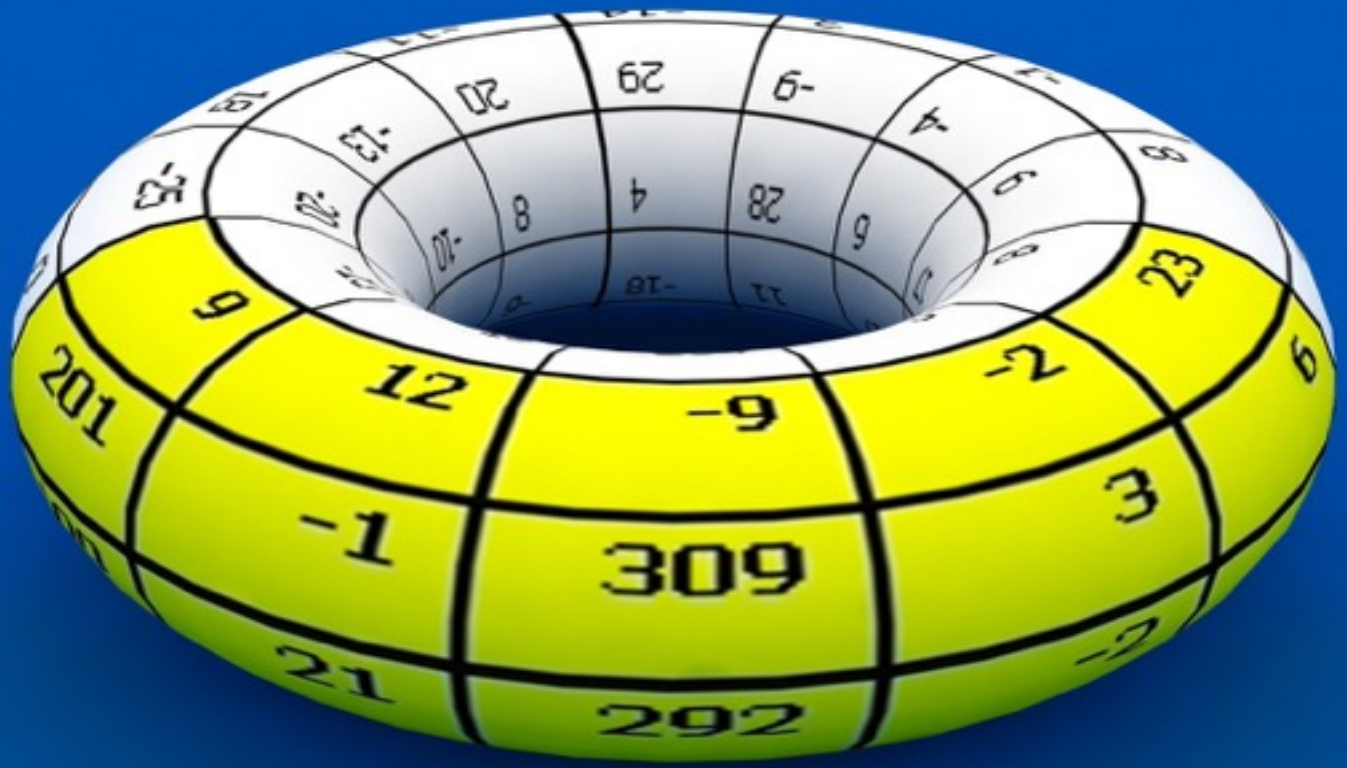
8	-10	-3	26	-11	-1	-6	12	17	6	28	4
20	-13	-20	-13	-15	-254	5	8	9	-4	-9	29
-11	18	−25	9	12	-9	-2	23	8	-1	3	-14
-16	-7	0	201	-1	309	3	6	−18	11	24	-8
-1	-7	11	100	21	292	-2	2	-18	-8	-10	9
26	-11	-19	-18	20	-981	2	-14	12	-14	1	27
9	-20	5	28	-15	26	-20	-8	-16	30	3	20
-6	-7	-5	-9	-16	-15	5	-16	22	-17	11	-18

扩展 2.5 维度

- 如果这个二维数组首尾相连，上下相连呢?
- 请花一分钟画出这个 2.5 维度的二维数组

- 倒计时 60 – 0 秒

扩展2.5D result



扩展：3D

- 一个立方体，每个（x，y，z）点有一个数值
- 如何求得最大子立方体，使它的和最大?

效能分析

- 请看教材 2.2 节

练习2：WordCounter program

- 教材 2.4.2 实践最简单的项目：WC
- 贯彻“做中学”的思想，动手实现下面的项目，并和别人的成绩相比较，分析产生差距的原因。
- 实现一个简单而完整的软件工具（源程序特征统计程序）。
- 进行单元测试、回归测试、效能测试，在实现上述程序的过程中使用相关的工具。
- 进行个人软件过程（PSP）的实践，逐步记录自己在每个软件工程环节花费的时间。
- 使用项目管理系统，练习使用其中的事件跟踪系统