

# 20180925 作业点评

教师对于学生作业的点评，不是对学生价值观、道德素养、品性的评判，也不是对学生本人的评判，只是出于教师对软件工程课程的理解，对学生当次作业的点评。无论对于评分的结果、评分的原则、评分所涉及到的价值观，都期待讨论。

初学者的表现不符合教师期待是正常现象，教师未达到学生期待的水平也是现状。正因为这样，才需要在“做中学”。希望我们能在对困惑的讨论中进步，无论教师还是学生。

教学上，整理总结的目的，是帮助同学们了解前车之鉴，以他山之石攻己之玉。同时，也避免在同一个坑里重复扣分。

## 目录

20180925 作业点评 .....	1
1. 扣分原因 .....	3
1.1. 没有预览 .....	3
1.2. 排版混乱 .....	3
1.3. 错别字 .....	5
1.4. 回答不完整 .....	7
1.5. 缺少证据 .....	11
1.6. 累积图错误 .....	12
1.7. 完成不精确 .....	13
1.8. 答案与问题不对应 .....	13
1.9. 发现了问题，然后呢？ .....	15
1.10. 拒绝作业 .....	15

1.11.	方法错误.....	17
1.12.	版本控制未 <b>build-clean</b> .....	18
1.13.	不是单元测试.....	19
2.	教师的感想.....	21
2.1.	计算错误.....	21
2.2.	两种记录对比.....	22
2.3.	不同量纲不应对比.....	24
2.4.	审题有误.....	26
2.5.	感谢.....	26
2.6.	沟通.....	27
2.7.	精确.....	28
2.8.	具体.....	28
2.9.	承诺.....	29
2.10.	工程技术不仅理论推测.....	30
2.11.	缺少过程.....	30
2.12.	进步还是妥协.....	32
2.13.	预览.....	34
2.14.	记录完整.....	34
2.15.	各种方法.....	35
2.16.	甘特图.....	38

# 1.扣分原因

## 1.1.没有预览

例 1

ΔT	净时间 ( $\Delta T$ )
	103min

教师因为看不懂“净时间 ( $\Delta T$ )”的含义而扣分。

## 1.2.排版混乱

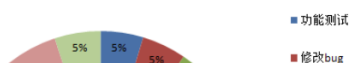
例 2

1.psp表格

Task (任务)	ST (开始时间)	ET (结束时间)	INT (中断时间)	ΔT(净时间)
功能测试	2018.10.1 9:47	2018.10.1 12:32	41	124
修改bug	2018.10.1 18:10	2018.10.1 20:38	19	129
效能分析	2018.10.2 13:26	2018.10.2 16:59	44	229
实现四则运算功能一及代码规范	2018.10.3 16:13	2018.10.3 20:47	48	226
实现四则运算功能一	2018.10.4 10:24	2018.10.4 16:17	61	292
实现四则运算功能二	2018.10.4 18:13	2018.10.4 21:07	16	158
实现四则运算功能二	2018.10.5 8:41	2018.10.5 14:09	42	286
实现四则运算功能三	2018.10.5 15:13	2018.10.5 17:24	14	117
实现四则运算功能四	2018.10.5 17:52	2018.10.5 20:49	14	169
实现四则运算功能四	2018.10.6 9:14	2018.10.6 18:27	61	492
单元测试	2018.10.7 10:29	2018.10.7 16:07	40	298
吉林市二日游	2018.10.7 18:23	2018.10.7 19:09	32	134

2.psp饼状图

PSP饼状图



教师因为看不懂红圈中的内容而扣分。

例 3

博客	规格书	2018.10.7/17:12	2018.10.7/18:58	0min	106min	本周进度条
编程	写四则运算	2018.10.7/19:01	2018.10.7/23:23	46min	215min	
编程	写四则运算	2018.10.8/7:35	2018.10.8/9:01	0min	86min	
博客	写博客	2018.10.8/9:03	2018.10.8/11:34	2min	149min	
博客	写博客	2018.10.8/15:18	2018.10.8/19:37	2min+10min	245min	
进度	代码行数	博文字数	所用知识点			
三、四周	686	2585	效能分析,功能测试,重定向			

教师因为看不懂红圈和红框中的内容而扣分。

例 4

项目	2018.10.06 14:43	2018.10.06 16:19	15+6	75										
博文	写博文	2018.10.06 14:43	2018.10.06 16:19	15+6	75	<table border="1"> <thead> <tr> <th>周数</th> <th>代码行</th> <th>博文字数</th> <th>用到的知识点</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>297</td> <td>5602</td> <td>UnitTest单元测试, 效能分析, 结对编程</td> </tr> </tbody> </table>	周数	代码行	博文字数	用到的知识点	3	297	5602	UnitTest单元测试, 效能分析, 结对编程
周数	代码行	博文字数	用到的知识点											
3	297	5602	UnitTest单元测试, 效能分析, 结对编程											
项目	编程完成功能一、二	2018.10.07 09:17	2018.10.07 12:10	19+25+7	122									
项目	编程完成功能三、四	2018.10.07 14:00	2018.10.07 17:29	34+5+8	167									
博文	写博文	2018.10.07 20:11	2018.10.07 22:01	18+15	77	<table border="1"> <thead> <tr> <th colspan="2">博文字数累积折线图</th> </tr> </thead> <tbody> <tr> <td>2450</td> <td>10215</td> </tr> </tbody> </table>	博文字数累积折线图		2450	10215				
博文字数累积折线图														
2450	10215													
项目	单元测试	2018.10.08 14:56	2018.10.08 16:33	11+15	71									
博文	写博文	2018.10.08 18:10	2018.10.08 19:35	13	72									
博文	写例行报告	2018.10.09 16:25	2018.10.09 17:43	8+6	64									

教师因为看不懂红圈中的内容而扣分。

例 5

51min	总时间：2079min	
114min		
103min		
83min		
163min		本周进度条
96min		

教师因为看不懂红框中的内容而扣分。

## 1.3. 错别字

例 6

git地址：<https://git.coding.net/Zy19951>

(1) 标题**技术时**没有将重复的单词排除。

以gongneng 1.txt为测试样本

实现截图：

教师根据上下文不能理解“技术时”的含义，因此扣分。

```
my      1
english 1
is      1
pool    1
```

预期结果：

作证材料：

```
C-A. 选择管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.17134.1]
(c) 2018 Microsoft Corporation. 保留
```

教师因为不知道什么是“作证材料”及与作业的关系，因此扣分。

例 7

要求，大致列出每个功能实现的方法，之后我就进行了加紧学习。接触到python才感受到它有多么强大，很容易上手，代码可读性非常高，索性python运用起来比较好上手，我也算及时赶上了进度，代码规范的时候，我们在python书写的基本规范上

应作“所幸”。

例 8

在讨论决定使用哪种方式实现的时候，实际操作后发现不正确，智能重新来过

应作“只能”。

## 1.4. 回答不完整

要求1 bug计分。阅读教材第298页，发布bug报告。

要求包括: (1)bug标题, (2)bug内容, 含测试环境和准备工作、测试的步骤、运行结果、期待的结果、运行结果与期待结果间的差异, (3)佐证材料, 至少包括运行截图、引用作业要求 (wf spec)中的条款。

1.段晓睿同学功能1下面多了一行

测试环境: Windows 7 控制台测试

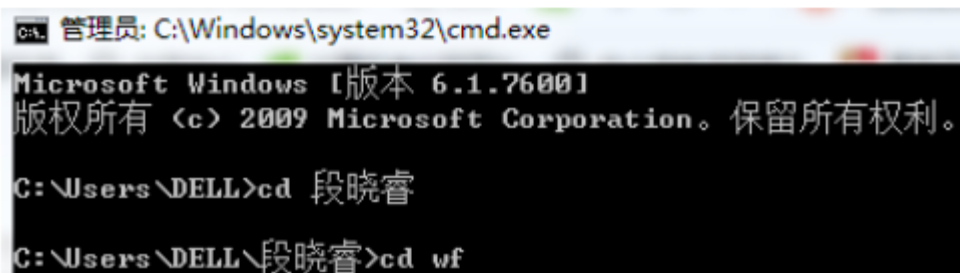
期待结果:

```
>type test.txt
My English is very very pool.

>wf -s test.txt
total 5

very    2
my      1
english 1
is      1
pool    1
```

运行结果:



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7600]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\DELL>cd 段晓睿
C:\Users\DELL\段晓睿>cd wf
```

教师寻找红框中标示的作业要求的项目困难, 文字和图示难以对应到要求, 因此扣分。

例 9

### 要求3：修改bug。

```
C:\Users\ws\Desktop\Word_Count\wf\wf3\wordlist\bin\Debug>wf folder

1.txt
Total: 47068 words

the: 27138
and: 13159
to: 14667
of: 13163
a: 8896
in: 7358
he: 7138
that: 6728
his: 6556
was: 6308

2.txt
Total: 106 words

the: 13
to: 8
is: 7
are: 6
for: 6
dead: 5
lanterns: 4
a: 4
their: 4
they: 3
```

git地址：[https://coding.net/u/Cool\\_Ary/p/Word\\_Count/git](https://coding.net/u/Cool_Ary/p/Word_Count/git)

标题、一张图、git 地址。教师看不懂是什么意思，为什么贴这么一张图，看不出与作业要求的关系，因此不计分。

例 10

要求包括：(1)bug标题，(2)bug内容，含测试环境和准备工作、测试的步骤、运行结果、期待的结果、运行结果与期待结果间的差异，(3)佐证材料，至少包括运行截图、引用作业要求(wf spec)中的条款。

环境：在windows下，运行wf.exe，支持命令行输入

1.刘佳瑞同学设置命令行参数

```
C:\Users\mac\Desktop\Ljr6899-wf.Java-origin\wf.Java-origin\wf>wf -s test.txt
C:\Users\mac\Desktop\Ljr6899-wf.Java-origin\wf.Java-origin\wf>
```

2.运行于洋同学的程序出现异常

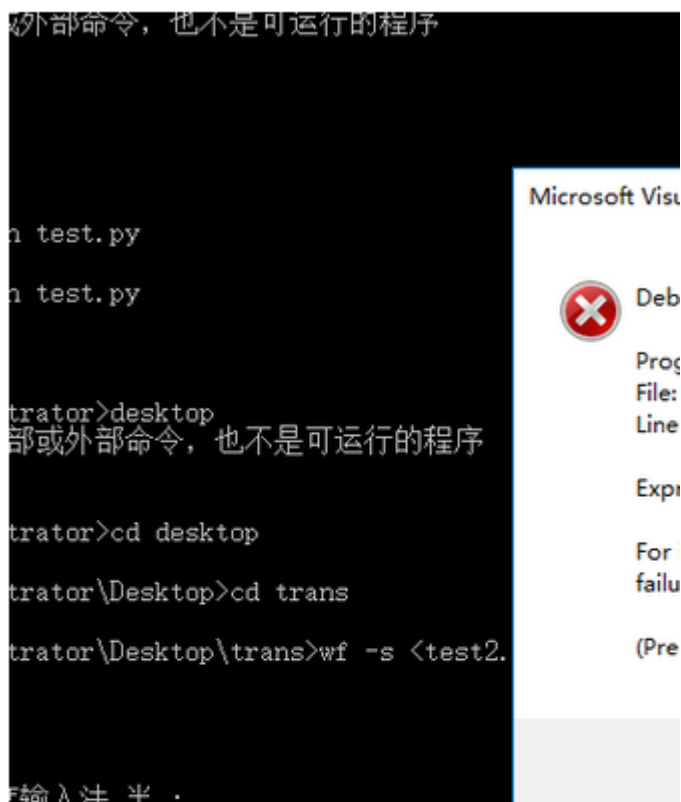
```
C:\Users\mac\Desktop\onion102983-cipintongji-master\cipintongji-master>wf -s war_and_peace
Traceback (most recent call last):
  File "wf.py", line 53, in <module>
    main(sys.argv[1:])
  File "wf.py", line 44, in main
    doCount(sys.argv[2])
  File "wf.py", line 21, in doCount
    words = re.findall(r'[a-z0-9^!@#$%^&*~\.\-]', f.read()).lower()
```



教师寻找红框中标示的作业要求的项目困难，文字和图示难以对应到要求，因此扣分。

例 11

### 1、周昊同学的不能运行



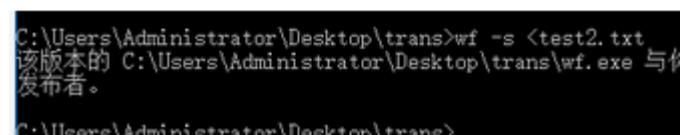
### 2、付佳同学

版本不兼容



### 3、于洋同学

版本不兼容



深切怀疑是不是我的电脑有问题

教师寻找作业要求中的项目困难，文字和图示难以对应到要求，因此扣分。

## 例 12

建立git : <https://git.coding.net/KamiForever/debug.git>

潘世维功能—能正 缺输出total

```
D:\homework\debug\潘世维>wf -s test.txt
total 5

very                2
english             1
my                  1
pool                1
is                   1
```

教师看不懂截图是改正过的，还是改正前的佐证材料，因此扣分。此题中的错别字或者语序错误，加重了教师的理解困难，对于扣分有贡献。

回答不完整的另一种例子，作业要求明确提及次数，要求量化。

要求2 记录完成本周作业过程中，《代码风格规范》执行情况。包括记录每一条规则的违反和纠正、**次数**、规范中某项条款的增强、消弱、或废除。在编码过程中完全未被违反的条款，说明过于宽松，或你们已经完全“本能”反应，不应作为条款，在本要求及要求1中不得分。此种情况，应补充一条新的条款。(20分)

但是同学的作业中模糊处理。

## 例 13

1.在给变量命名时，我们出现**小部分没有意义**的变量，由于不能让每个变量都具有实际意义，在实现功能四的时候，我们程序里出现了**几个没有意义**的变量名，其他函数里也出现了个别没有意义的变量（对应9命名）

## 例 14

(2) 定义的函数的代码行数不宜过长，10行左右最佳。同样是增加代码的可读性。但是我们的能力有限，**有几个函数都超过了10行**。

## 例 15

回答不完整的，还包括在吉林市二日游作业中，有同学缺少 风险管理、人员分工、经费、甘特图等重要指标。因为每个子作业分数有上限，所以无论其他部分做得非常好，如果缺项必然导致扣分。避免缺项的简单方法，就是小学老师在写作文时要求的“列大纲”。

## 例 16

2) 火车：( 人数统计截止到2018年10月1日 20 : 00 )

自行买票，买不到的就要坐大巴

推荐购买城际高铁：G1005，G1201，G1203。

这样的方案，可能不少同学会评价为“不好”。但是如果缺少这一项目(或者没有提到具体方案)，那么就是“差”，因为无从评价，因此无法开始讨论和改进。

例 17

付佳同学的作业可以作为模板参考。

在这里[<http://www.cnblogs.com/fuj905/p/9755652.html>]。

## 1.5. 缺少证据

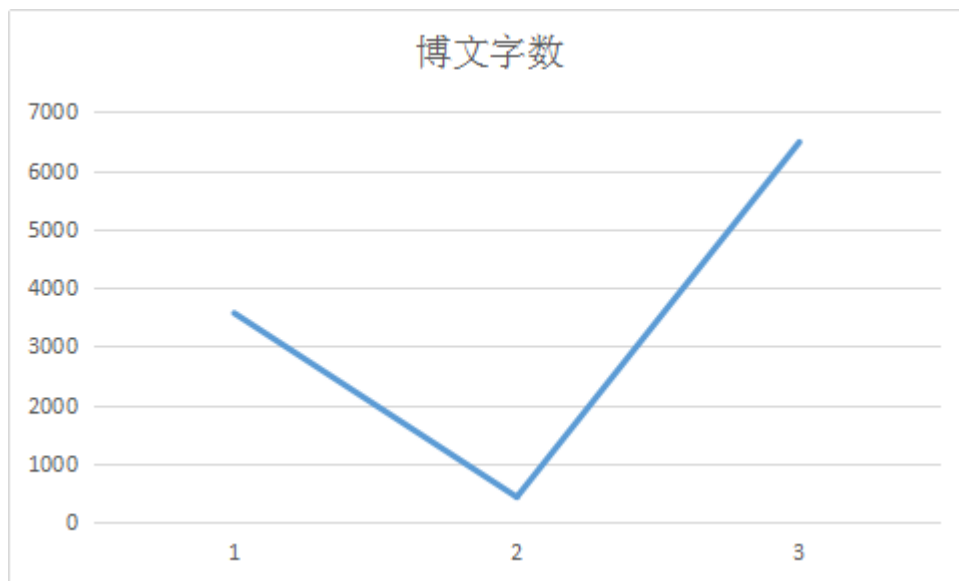
例 18

自己在重新为自己代码找bug时电脑全程崩溃，把程序换到另一个盘里也是全程崩溃，不好用

，但是在完成测时间的作业之后，执行新程序的时候全程电脑崩溃掉了，所以没法提供执行截图。

这是个经典的现象，称为“我的代码被猫吃了”。就这个案例而言，容易解决。代码还在 git 上，另找台电脑运行截图就是。如果运行你的程序就会导致电脑崩溃，那么是个很好的学习机会，解决这个问题。如果最后的一段代码没有版本控制，重写就是。没有证据，只能扣分。

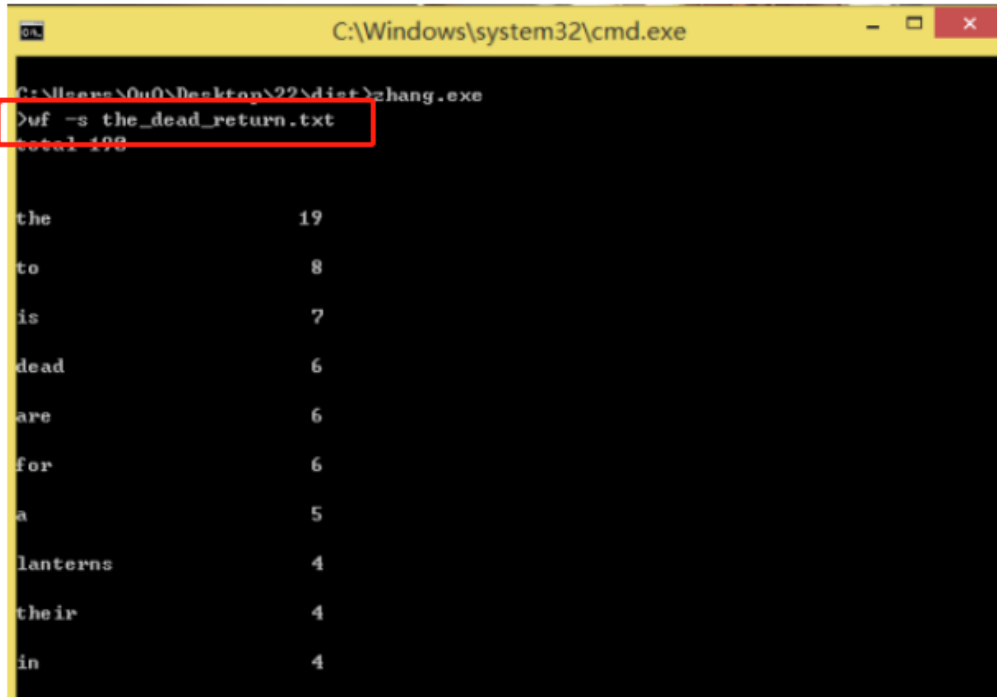
## 1.6. 累积图错误



为什么是错的，参见前两次教师对作业的点评。在工作中，第一次错，上级会认为你是无知是基础不好；第二次错，是学习能力较差或者接受新知识较慢；第三次错，是态度问题。

## 1.7. 完成不精确

测试书籍利用程序统计单词数与在WPS中统计的个数不符。



```
C:\Windows\system32\cmd.exe
C:\Users\0u0\Desktop\22\dist>zhang.exe
>uf -s the_dead_return.txt
total 198
the 19
to 8
is 7
dead 6
are 6
for 6
a 5
lanterns 4
their 4
in 4
```

红框中，这不是命令行参数，而是在程序中通过控制台输入。此题目要求是 bug 修改，所答暴露有重大 bug 未修正，不符合要求，因此扣分。

## 1.8. 答案与问题不对应

例 19

“下图分别为优化后的运行时间与在要求 1 中最花费时间的 3 个函数在优化后的运行时间。”

13	0.000	0.000	0.000	0.000	functools.py:73(wraps)
1	0.056	0.056	17.944	17.944	fuf.py:1<<module>>
1	0.185	0.185	0.266	0.266	fuf.py:12(p_count)
20165	0.001	0.000	0.001	0.000	fuf.py:17<<lambda>>
1	0.014	0.014	0.309	0.309	fuf.py:4(getText)
9/1	0.002	0.000	17.944	17.944	<built-in method builtins.exec>
148	0.000	0.000	0.000	0.000	<built-in method builtinsgetattr>
54	0.000	0.000	0.000	0.000	<built-in method builtins.hasattr>
1	17.182	17.182	17.182	17.182	<built-in method builtins.input>
52	0.000	0.000	0.000	0.000	<built-in method builtins.isinstance>

教师找不到与问题的对应关系，即所述的 3 个函数(也因此看不到优化修改的效果)，因此扣分。

例 20

要求4

```
ca. 管理员: C:\Windows\system32\cmd.exe
that 8087
his 7966
was 7327

36 function calls in 1.228 seconds

Ordered by: internal time

ncalls  tottime  percall  cumtime  percall  filename:lineno(function)
   1    0.560    0.560    1.228    1.228  wf.py:2(<module>)
   1    0.203    0.203    0.203    0.203  <method 'split' of 'str' objects>
  15    0.143    0.010    0.143    0.010  <method 'replace' of 'str' objects>
)
   1    0.114    0.114    0.114    0.114  <method 'read' of '_io.TextIOWrapp
er' objects>
   1    0.107    0.107    0.107    0.107  <method 'lower' of 'str' objects>
   1    0.054    0.054    0.054    0.054  <method 'sort' of 'list' objects>
   1    0.034    0.034    0.034    0.034  wf.py:18(<listcomp>)
  11    0.012    0.001    0.012    0.001  <built-in method builtins.print>
   1    0.000    0.000    1.228    1.228  <built-in method builtins.exec>
   1    0.000    0.000    0.000    0.000  <method 'items' of 'dict' objects>

   1    0.000    0.000    0.000    0.000  <built-in method builtins.len>
   1    0.000    0.000    0.000    0.000  <method 'disable' of '_lsprof.Prof
iler' objects>
```

教师找不到与问题的对应关系(也因此看不到优化修改的效果)，因此扣分。

例 21

```
命令提示符
away 1
until 1
can 1
seen 1
no 1
more 1

281 function calls (265 primitive calls) in 0.025 seconds

Ordered by: internal time

ncalls  tottime  percall  cumtime  percall  filename:lineno(function)
 109    0.024    0.000    0.024    0.000  {built-in method builtins.print}
   2    0.000    0.000    0.000    0.000  {built-in method io.open}
   1    0.000    0.000    0.024    0.024  wf.py:12(count_words)
   1    0.000    0.000    0.000    0.000  {method 'findall' of 're.Pattern' objects}
   1    0.000    0.000    0.000    0.000  {method 'read' of '_io.TextIOWrapper' objects}
   1    0.000    0.000    0.025    0.025  wf.py:1(<module>)
   1    0.000    0.000    0.000    0.000  {method 'readlines' of 'io._IOBase' objects}
   1    0.000    0.000    0.000    0.000  {built-in method _collections._count_elements}
   1    0.000    0.000    0.000    0.000  {built-in method builtins.sorted}
  7/1    0.000    0.000    0.000    0.000  {built-in method _abc._abc_subclasscheck}
   1    0.000    0.000    0.000    0.000  sre_parse.py:475(_parse)
   3    0.000    0.000    0.000    0.000  {method 'split' of 'str' objects}
   1    0.000    0.000    0.000    0.000  wf.py:5(counts)
   1    0.000    0.000    0.000    0.000  re.py:271(_compile)
   1    0.000    0.000    0.000    0.000  {built-in method _abc._abc_instancecheck}
   2    0.000    0.000    0.000    0.000  {built-in method _locale._getdefaultlocale}
  2/1    0.000    0.000    0.000    0.000  sre_compile.py:71(_compile)
```

没错，profile 是这样的，但是教师读不懂哪里是最耗费时间的 3 个函数，因此扣分。

代码规范作业也有同学把问题 1 规范和问题 2 过程记录混在一起。因为教师努力看懂了，所以

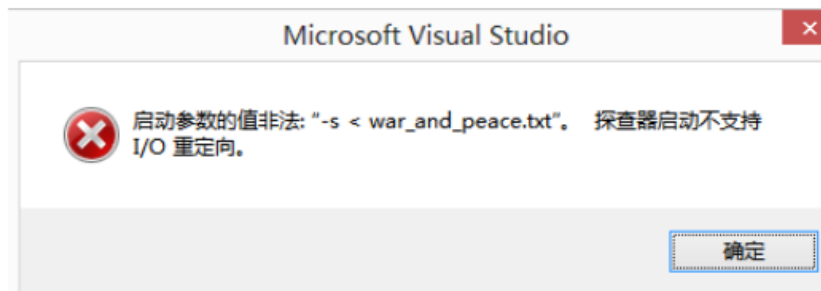
没有扣分。

## 1.9. 发现了问题，然后呢？

例 22

1. 以 战争与和平 作为输入文件，重读向由文件系统读入。

由于在vs中使用profile测试功能4会出现如下错误：



多次安装vs2015也并没能完成测试，最后CPU参数的测定以及后续的测试使用了very-sleepy。

以前的同一作业中，有同学给出了解决方法。

## 1.10. 拒绝作业

例 23

要求1：我运用乔静玉同学的代码，我认为运行的速度较快，认为不用优化，所以无法给出修改。

要求2：根据老师给的作业参考资料，在cmd中输入以下命令

```
1 python -m cProfile wf.py war_and_peace.txt
```

得到分析结果如图：

```
C:\Users\OuO\Desktop\借用的代码效能测试>python -m cProfile wf.py war_and_peace.txt
Traceback (most recent call last):
  File "E:\Python\lib\runpy.py", line 170, in _run_module_as_main
    "__main__", mod_spec)
  File "E:\Python\lib\runpy.py", line 85, in _run_code
    exec(code, run_globals)
  File "E:\Python\lib\cProfile.py", line 160, in <module>
    main()
  File "E:\Python\lib\cProfile.py", line 145, in main
    with open(prognane, 'rb') as fp:
FileNotFoundError: [Errno 2] No such file or directory: 'wf.py'

C:\Users\OuO\Desktop\借用的代码效能测试>
```

我认为是代码出现了错误，但由于是借用进行测试的代码，首先我没有做出这两个功能，其次我没有能力进行改正。

因为拒绝(括号里有情有缘的理由和谦和的态度会被忽略——这位同学判断不必优化、没有能力改正——拒绝)完成作业，因此扣分。

有的同学可能会说，因为我用的是 python，所以没法 profile 吧。或者有的同学说，我都贴图了，老师你看不明白么？以下是优秀示范，教师可以读懂。

其中他/她特意在文字提到，哪几个函数、这几个函数是做什么的、它们在优化前的耗时、它们在优化后的耗时、因为优化整个程序的时间消耗减少了多少。非常清晰。

例 24

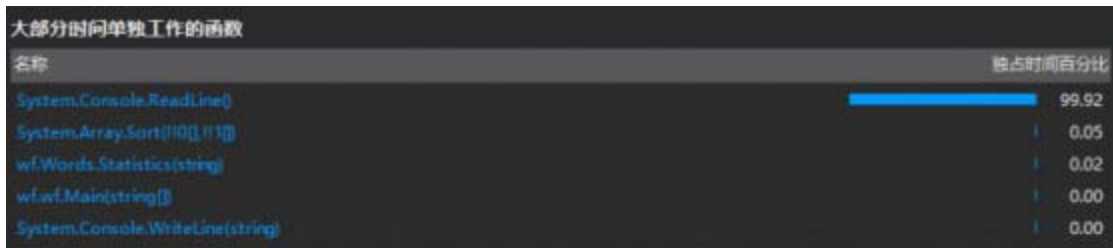
[<http://www.cnblogs.com/fuj905/p/9737359.html>]

例 25

[<http://www.cnblogs.com/silentteller/p/9747445.html>]

例 26

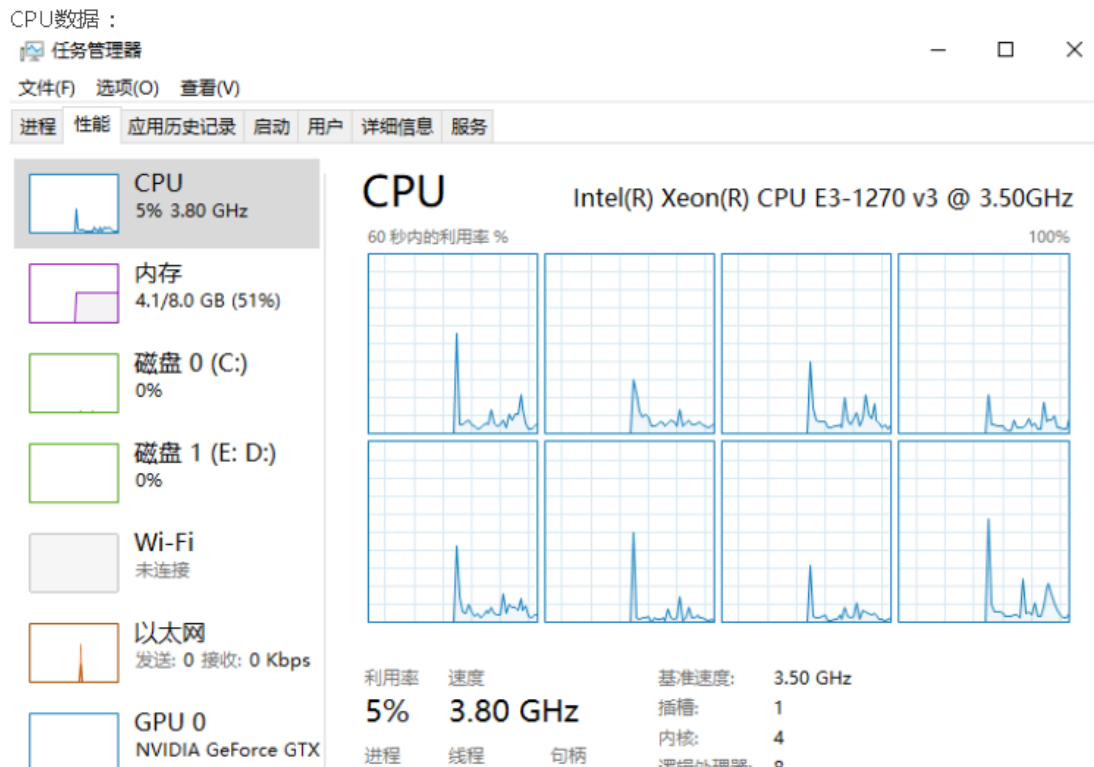




这位同学已经 profile 到了最耗费时间的函数 ReadLine，换个函数读入可能速度就能提高，但是他没有继续往下做。

## 1.11. 方法错误

例 27



这不是 profile，并且不符合下述作业要求。

要求2 通过 profile 找出程序的瓶颈。给出程序运行中最花费时间的3个函数(或代码片断)。要求包括截图。(5分)

要求 分析为什么此处是瓶颈。  
 要求 profile 需要得到函数的运行时间和次数。仅得到CPU和内存占用，不得分。

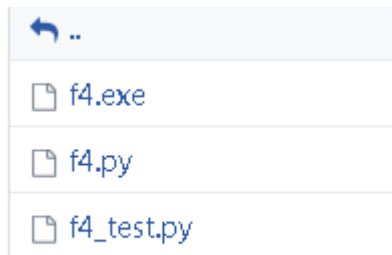
## 1.12. 版本控制未 build-clean

### 作业要求

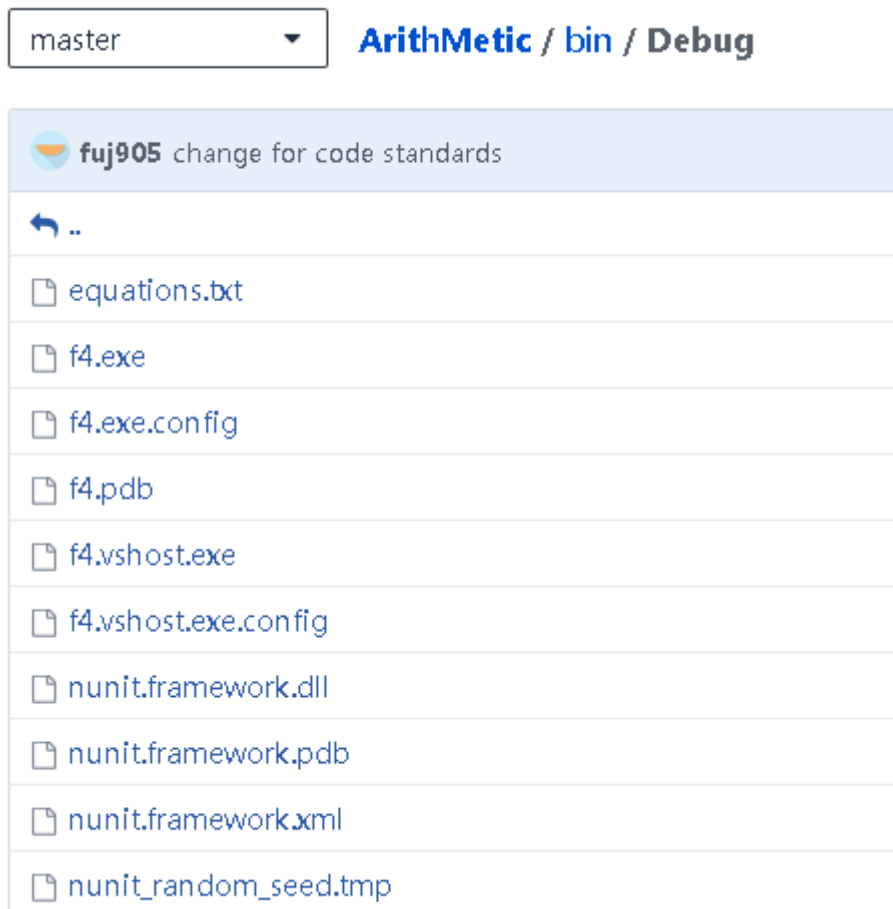
要求4 使用coding.net做版本控制。checkin 前要求清理 临时文件、可执行程序，通常执行 build-clean可以达到效果。（5分）

### 例 28

未 build-clean，版本控制中包括 exe 文件，因此不符合要求。扣分。



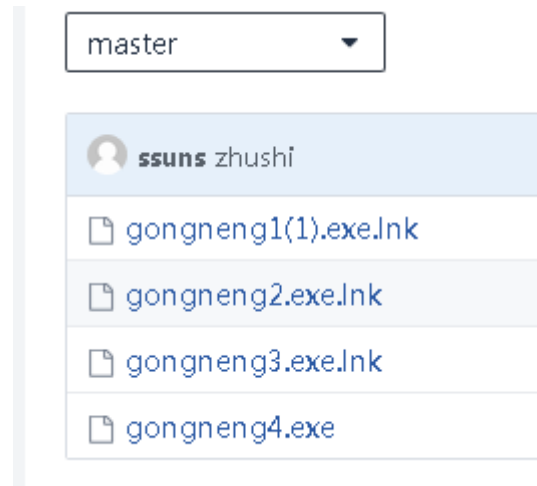
### 例 29



未 build-clean，版本控制中包括 exe 等文件，因此不符合要求。扣分。

还有完全没有源代码的。

例 30



## 1.13. 不是单元测试

作业要求使用 TDD。TDD 是术语。TDD 测试不是针对功能的(从程序外部可见)，而是针对模块(类和函数)的。

2人结对，使用TDD测试框架(如JUnit, JUnit, cppUnit等)完成本周作业四则运算试题生成的单元测试。

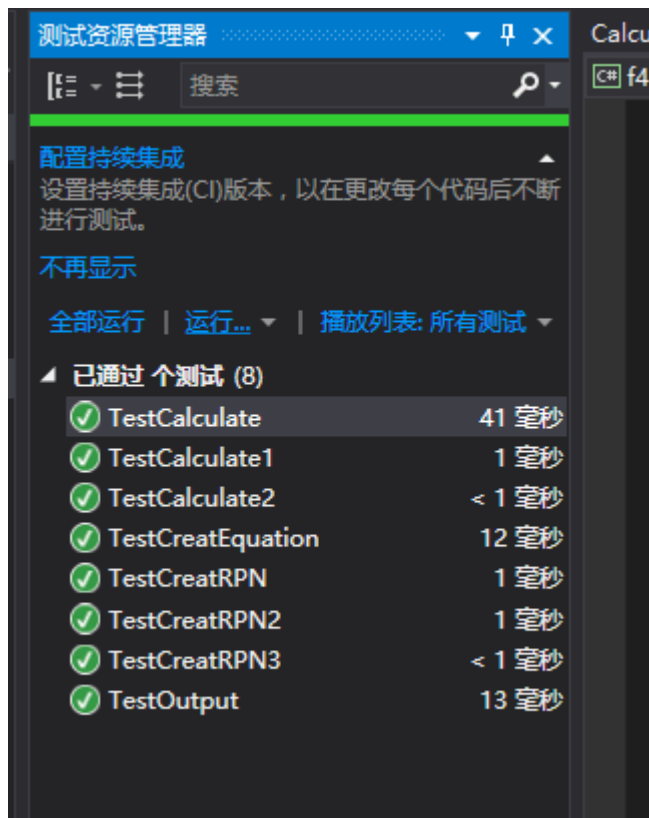
例 31

## 测试用例

ID	测试用例	预计结果
1	用户输入f4	出现表达式及文号
2	用户输入非f4	**不是内部或外部命令，也不是可运行的程序或批处理文件。
3	用户输入f4后对于表达式做出正确答案	打印：“答对啦，你真是个天才!”并给出下一题及换行后的问号
4	用户输入f4后对于表达式做出错误答案	打印：“再想想吧，答案似乎是12喔!”并给出下一题及换行后的问号
5	用户输入f4后并答完20道题，其中答对*道题	打印：“你一共答对*道题，共20道题”
6	屏幕提示 $5+15/3+5 =$	15

TDD 是一种自动化测试手段，需要用肉眼观察的，不符合要求。应该有如下过程。

例 32



有的同学则由错误走向正确，在学习中确有收获。

### 例 33

在最开始时我们对于单元测试的理解有误认为只要能运行便是正确的，于是我们最开始写的单元测试只是简单的调用程序看是否出结果，写出了以下一段代码。

```
print(-----开始测试-----\n)\nprint(请输入f4开始测试功能1)\nimport gongneng1\nprint(\n-----功能1测试结果-----)\nprint(请输入f4开始测试功能2)\nimport gongneng2\nprint(\n-----功能2测试结果-----)\nprint(请输入f4 -c x (x为要显示题目的数量)开始测试功能3)\nimport gongneng3\nprint(\n-----功能3测试结果-----)\nprint(请输入f4 -c x (x为要显示题目的数量)开始测试功能4)\nimport gongneng4\nprint(\n-----功能4测试结果-----)\nprint(\n-----所有测试结果-----)
```

但在观看往届同学的单元测试发现，单元测试是指对软件中的最小可测试单元进行检查和验证并非是运行整个程序。在学习宋雨雨同学的单元测试博文后，我们再次尝试进行单元测试。

功能1的单元测试代码：

```
import unittest\nfrom gongneng1 import *\n\nclass TestDict(unittest.TestCase):\n\n    def test_calc(self):\n        print(-----开始测试-----\n)\n        str = ra_number()\n        print(str)\n        result = input(输入正确结果\n>)\n        self.assertEqual(int(result), my_eval(str))\n        print(\n----- 测试结果-----)
```

## 2.教师的感想

### 2.1.计算错误

#### 例 34

类型 ( Type )	任务 ( Task )	开始时间 ( ST )	结束时间 ( ET )	中断时间 ( INT )	净时间 ( ΔT )
结对编程	结对编写功能一代码	2018.10.3 16:00	2018.10.3 23:49	188min	581min

16:00 至 24:00，8 小时合 480 分钟。

如何避免这种错误呢？有很多办法，比如交叉检查，比如画图(chart)，比如找离异值然后复核。

## 2.2. 两种记录对比

例 35

日期	内容	ST (开始时间)	ET (结束时间)	INT (中断时间)	T (净时间)
2018.10.1	写吉林两日游	19:27	20:43	4+8	64min
2018.10.2	写功能测试	20:09	22:53	17+7	140min
2018.10.3	写效能分析作业	9:11	15:32	4+17+6+7+11	336min
2018.10.3	四则运算功能1及代码规范	16:13	20:47	13+9+4+5+9+8	226min
2018.10.4	四则运算功能1	10:24	16:17	24+13+11+7+6	292min
2018.10.4	四则运算功能2	18:13	21:07	4+5+7	158min
2018.10.5	四则运算功能2	8:41	14:09	12+4+7+19	286min
2018.10.5	四则运算功能3	15:13	17:24	4+3+7	117min
2018.10.5	四则运算功能4	17:52	20:49	7+4+3	169min
2018.10.6	四则运算功能4	9:14	18:27	3+7+18+3+6+5+11+	492min
2018.10.7	单元测试	10:29	16:07	19+4+7+2+8	298min
2018.10.7	写四则运算感受	18:32	19:09		0 37min
2018.10.8	写例行报告	17:27	19:06	8+9	82min

上图优于下图的记录。记录首要原则是保留事实，如有可能保留尽可能多的事实。单次记录 20 多分钟，如果不是持续 20 多分钟的话，那么，就是事后补记的。补记时常有误差和错误。具体和细节优于抽象和总结。

设计	四则运算软件设计	2018.9.26 20:45	2018.9.26 22:04	3min	76min
代码规范	制定代码规范	2018.9.27 12:28	2018.9.27 13:00	0	32min
编码	编码四则运算功能一	2018.9.27 21:24	2018.9.27 23:30	21min	105min
测试	词频统计功能测试	2018.9.28 20:53	2018.9.28 22:34	17min	84min
阅读	阅读《C#编程》	2018.9.29 19:47	2018.9.29 21:50	32min	91min
测试	词频统计功能测试	2018.10.1 14:12	2018.10.1 15:10	0	58min
阅读	阅读《C#编程》	2018.10.2 19:29	2018.10.2 21:19	7min	103min

下图这是个可行的方案，未来能够回忆起更多细节。

例 36

查询吉林市信息， 写规划书	2018.9.26 8:00	2018.9.26 12:48	56 洗漱，吃饭	292
------------------	-------------------	--------------------	----------	-----

下图也有利于阅读。

例 37

中断时长	净时
4+15=19	:
8+2=10	:
3+12+4=19	:
1+13=14	:
32+6=38	:
5	:
11+3=14	:
7+6=13	:
2+15=17	:
13	:
8	:
13+19=32	:
15	:
20	:
12	:

例 38

结束时间	中断时间	Delta时间
4 15 : 23	4min+2min	68min
4 16 : 14	0min	37min
4 20 : 32	17min+32min+5min+ 11min+14min	117min
4 22 : 25	9min+2min+5min	65min
5 11 : 58	0min	11min

颗粒度大导致失控或者不可跟踪，是个普遍问题。如

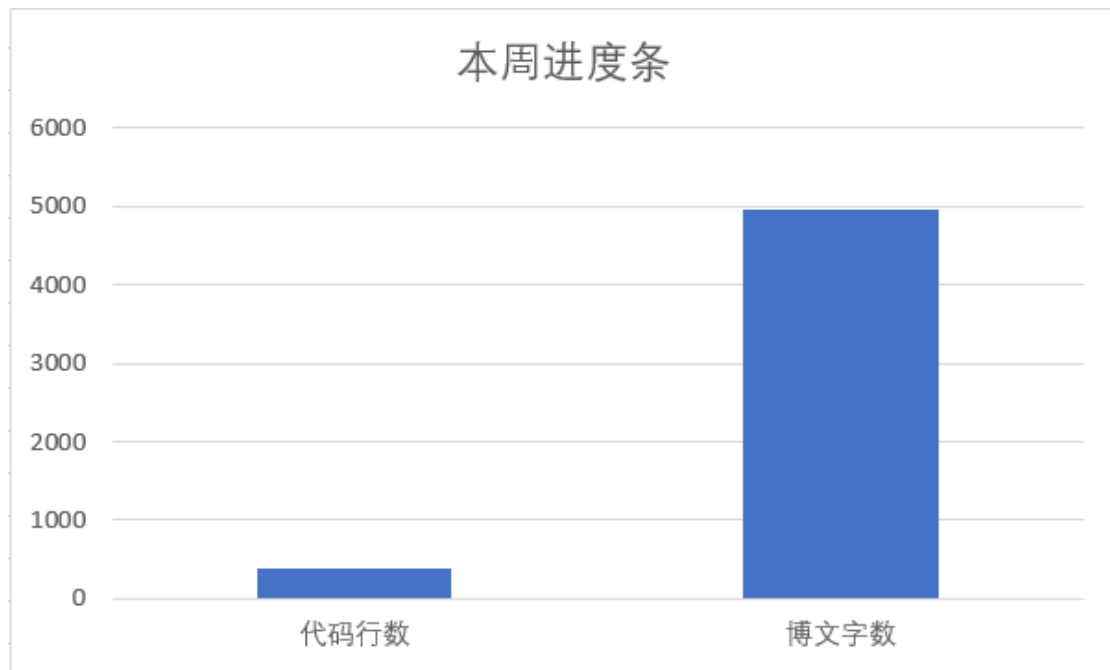
例 39

代码行数	397
博文字数	6482
知识点	单元测试、Java

Java 作为一个知识点的话，过于庞大。

## 2.3. 不同量纲不应对比

例 40



代码行数与博文字数，对比没有意义。

例 41



```

D:\aaaa\wc\wc_1\dist>ptime wf -s war_and_peace.txt
ptime 1.0 for Win32, Freeware - http://www.pc-tools.net
Copyright(C) 2002, Jem Berkes <jberkes@pc-tools.net>

=== wf -s war_and_peace.txt ===
total 566076
the      34423
and      21618
to       16553
of       14921
a        10466
he       9395
in       8631
his      7933
that    7475
was     7206

Execution time: 2.427 s

```

```

D:\aaaa\wc\wc_1>python -m cProfile -s time wf.py -s war_and_peace.txt
total 566182
the      34257
and      21392
to       16500
of       14905
a        10405
he       9297
in       8606
his      7930
that    7411
was     7200
      264303 function calls (264239 primitive calls) in 0.996 seconds

Ordered by: internal time

ncalls  tottime  percall  cumtime  percall  filename:lineno(function)
 65325   0.423   0.000   0.611   0.000  wf.py:38(countLine)
     1   0.180   0.180   0.964   0.964  wf.py:51(mains)
     4   0.108   0.027   0.108   0.027  {method 'sort' of 'list' objects}
 65325   0.099   0.000   0.099   0.000  {method 'split' of 'str' objects}

```

三次测试的时间都都在1s以下，相较于第一次的1.071有改善

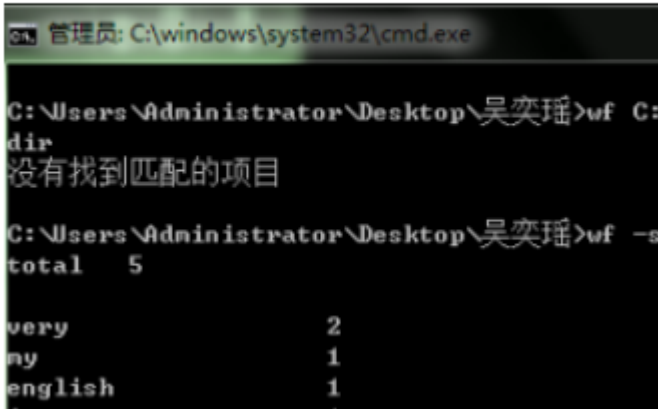
countline下降0.01s左右。

这位同学的“第一次”使用 ptime 测量时间，优化后的“三次测试”则使用 python -m cProfile -s time 测量时间。两种方法的结果不可以直接比较。

类似的，有的同学提到，他的程序速度较其他同学慢。如果不是在同一台计算机(软硬件配置相同)上运行，那么，同样不能直接比较。

## 2.4. 审题有误

5.标题：功能4没有实现



```
管理员: C:\windows\system32\cmd.exe
C:\Users\Administrator\Desktop\吴奕瑶>wf C:
dir
没有找到匹配的项目
C:\Users\Administrator\Desktop\吴奕瑶>wf -s
total 5
very          2
my            1
english      1
```

作业要求

- bug的颗粒度低于作业要求中的“功能”。作业要求共4个功能，其中某一功能未实现，不视为bug。

## 2.5. 感谢

例 42

### 1.1 bug标题：

于洋同学帮我找到功能三中单词总数输出格式的错误，缺少“words”

这位同学能够认识到，于洋同学找他的 bug 是“帮”。这些纠错，包括如何在工程中的严谨精神，以后都是工资，理应感谢，而不是怨恨。当出现问题，遮遮掩掩，就是这样：如果你的上级工程师不差评你，那么公司就会差评他，如果公司不差评价你们，那么用户就会差评公司和产品。

例 43

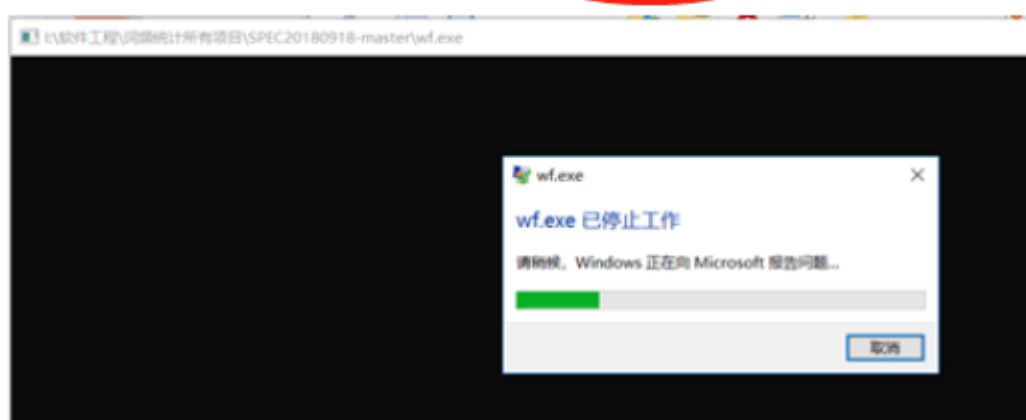
由于此课程作业的性质，导致本篇博客中的内容有对部分同学的代码进行测试，也有说错的地方，如果冒犯到你的话，我在这里提前和冒犯到的同学说声抱歉，如果有需要的话，我也可以当面向你道歉。

这样客气一下作为社交礼仪也许未偿不可，不过如果自己被挑 bug 的时候认为是冒犯，那么还需继续加强对自然科学的修行。

## 2.6. 沟通

例 44

2. 下载潘世维同学的文件也是无法运行，我开始怀疑是自己电脑的问题。



例 45

2、付佳同学

版本不兼容

```
C:\Users\Administrator\Desktop\trans>wf -s <test2
该版本的 C:\Users\Administrator\Desktop\trans\wf.
发布者。
C:\Users\Administrator\Desktop\trans>
```

3、于洋同学

版本不兼容

```
C:\Users\Administrator\Desktop\trans>wf -s <test2
该版本的 C:\Users\Administrator\Desktop\trans\wf.
发布者。
C:\Users\Administrator\Desktop\trans>
```

深切怀疑是不是我的电脑有问题

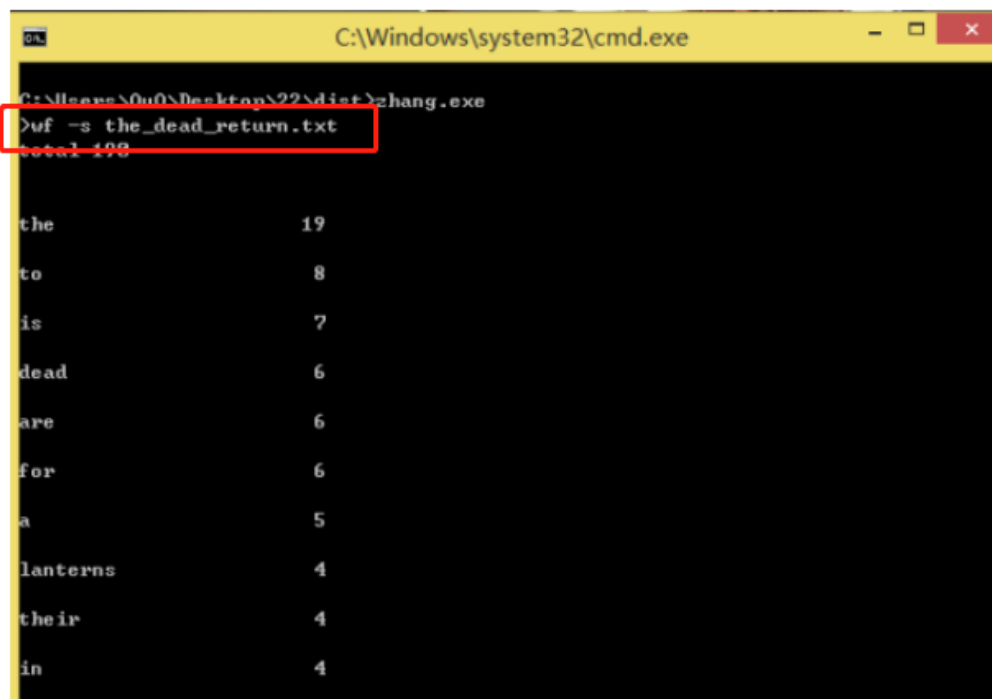
此时，应该去找程序员本人，请他运行，查看他与你运行的差异。这也是作业中要求下述内容的原因。

⑤, (2)bug内容, 含测试环境和准备工作、测试的步骤、运行结果、期待的结果、运行结果与期待结果间的差异, (具体要求 (wif.cnc)中的条款

不然怎么办? 把问题留给领导、上级工程师、教师、师兄? 对是否沟通如果有犹豫, 原则是过量沟通。

## 2.7. 精确

测试书籍利用程序统计单词数与在WPS中统计的个数不符。



```
C:\Windows\system32\cmd.exe
C:\Users\0u0\Desktop\22\dict\>zhang.exe
Duf -s the_dead_return.txt
total: 198

the          19
to           8
is           7
dead         6
are          6
for          6
a            5
lanterns    4
their       4
in           4
```

红圈中, 作业要求是 WORD, WORD 与 WPS 不是同一软件, 连拼写都不一样。

## 2.8. 具体

例 46

## 5. 潘世维功能—与实际想要结果不符

预期结果：

**功能1** 小文件输入。为表明程序能跑，结果真实而不是迫害老五盘在控制台输入命令。

```
>type test.txt
My English is very very pool.

>wf -s test.txt
total 5

very 2
my 1
english 1
is 1
pool 1
```

运行结果：

```
D:\homework\other\潘世维>wf -s test.txt
total 6

very 2
my 1
is 1
english 1
pool 1
```

上图中的红圈和箭头是教师加的。只摆在图在那里的态度就是，“你看吧，这不是很明显么。

“到底哪里不一样呢，具体地指出来，文档更有价值。

## 2.9. 承诺

例 47

由于时间问题，遗憾单词统计个数的bug和输出格式不正确的bug未能解决。争取日后修复这些bug。

## 例 48

名称	独占样本数百分比
isalpha	100.00
mainCRTStartup	0.00
__tmainCRTStartup	0.00
main	0.00
console_get_single	0.00

至于优化，我实在想不出来该如何进行优化，对此 我需要更深入的学习。

类似的还有“暂未实现”“以后加强”，我特别希望把同学们对未来的承诺都收集起来，以后逐一询问。不过只是希望，只是想想，和所有并不把这种承诺当回事的同学一样，只是表达成年人世界的礼貌地拒绝而已。

## 2.10. 工程技术不仅理论推测

### 例 49

由于程序使用了链表，在存储单词的过程中十分耗时，应该改用数组会更有效率，但是由于所需改动太大，没能实现。

通过这次效能测试，深刻的体会到了链表有多费时。

这位同学并未实现数组，因此不能以实证 profile 验证数组比链表更快。所以，是不应该有“深刻的体会的”。工程技术要求不仅理论推荐，还必须有实证的测量。

## 2.11. 缺少过程

作业要求报告过程。

**要求2** 在博客报告测试用例全部fail到全部pass的过程，报告事实（fail到修改代码或者测试用例，到pass）以及收获。除了最初的框架，测试用例中存在一次性pass没有经过fail的，也报告一次性通过，给出如此优秀地实现了这部分功能的代码。（40分）

有部分同学没有失败的过程，看起来测试一次性成功。基于大部分同学的经验，这是不可能的，在学业上也错过了学习和实践 TDD 过程的机会。但是，确实可能存在一次性通过的微小可能，并且作业没有要求必须有 fail，因此这种情况没有扣分。

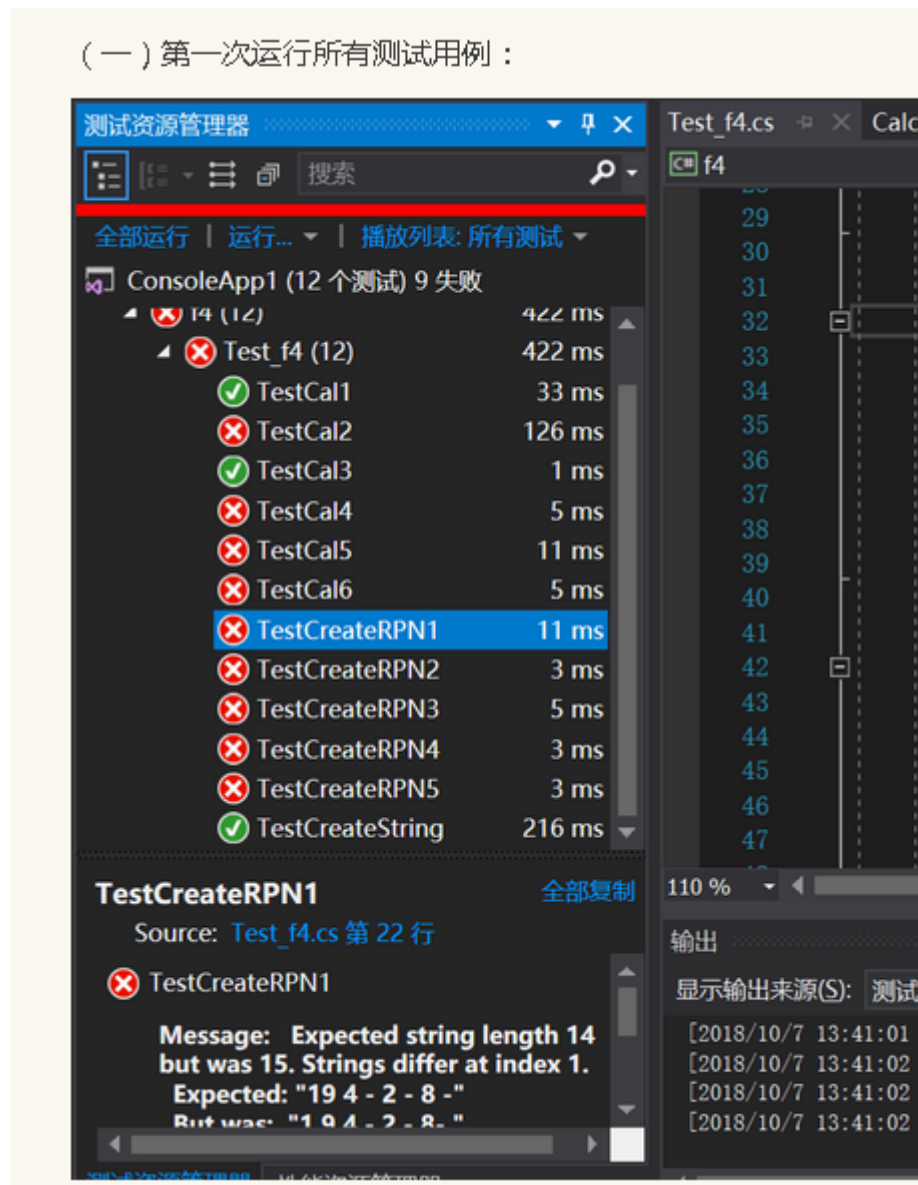
优秀作业 张俊余，有测试用例，有过程。

例 50

[<http://www.cnblogs.com/z1174299705/p/9753446.html>]。

另一位有过程的同学吴奕瑶的工作也非常好，是这样的。

例 51



因为有了过程，所以对于下述作法，教师就可以根据自己的经验评论。

例 52

```
public void TestRandNumber()
{
    int i = t.RandNumber();
    if (i > 0 & i < 20)
        Assert.Pass();
    else
        Assert.Fail();
}
```

通常，不用 if 判断，而是对方框中的内容为真作 assert。

## 2.12. 进步还是妥协

在代码规范中，有同学选择了消弱规范。如

例 53

4. 变量命名：用变量的小写单词命名。为了方便，允许少量的“a”，“b”，“m”，“n”存在，但不超过12个。

改为

变量命名	7. 消弱，允许使用更多单字母命名变量
------	---------------------

在当前项目规模的情况下，有超过地 12 个变量使用单字母命名变量，编程风格值得改进。但是，进步和代码控制能力并非一朝一夕的事情，在这种情况下，暂时妥协，降低难度，从而专注于训练工程管理，不失为可行之策。

例 54

这块我感觉是整个项目中最难处理的一部分，在网上搜了很多方法，比如树的最小表示法区分同构的树等等，我们尝试了很多但是没有达到预想的效果，所以最后我们两个人经过讨论决定使用这样一种判断：如果两条题目的结果相同、操作数和操作符相同（顺序可能不同，经过排序后比较是否相同），虽然没有非常精确地算法，但是输出的同样是没有重复的题目，同样完成了功能，因为结果不同的题目构成肯定不相同。缺憾就是生成过程中会多丢弃几个题目，但是对整体影响非常小，我们也对自己的想法比较满意；

这是不错的尝试。虽然并不完全是期待的效果，但是测试用例可能并未覆盖。也是不错的妥协。



### 例 55

这次作业本来我与[ ]同学是准备用C++来写的，但是在第一个功能时，我俩就遇到了问题，我们写的代码在运行时出现了三个运算符一样的问题（即第一个运算符是什么，后面两个运算符也是什么），这明显不符合题目要求，随后我俩开始改正，在改正的过程中我俩发生过争论，但是谁都不服谁，最后改用Python来实现这个作业。

首先，我们接触过Python，这是我第一次用，我又用了三天时间来学了一下Python，最终决定，王-可同学来编写，我指挥。在编写过程中，我们又因是否对函数添加注释产生了分歧，最终达成一致—添加注释；我们又对循环中是使用switch case还是if来进行条件判断产生分歧，最终决定使用if语句；我们对函数的命名规则产生分歧，我们的编写代码的时间不一致，总是错过，磨合了好多次，最后，我们的代码参考了学长的代码。

根据我的经验，三个运算符相同这种现象，不应该通过转换为 python 语言解决。这是逻辑问题，不是语言特性问题。

至于把抄袭代码(按课堂的定义，你看完重写一遍，重写时不参考就不算抄袭)表述为“参考”，一点儿也不能改变错误的性质。这种妥协，不会有一丁点儿进步。

### 例 56

功能：加分数。此方法借鉴了网上的其他大神的想法，就是随机生成浮点数，再用Python中limit\_denominator(10)函数强制转换为分母不大于10的分数，代码：

这种，教师难以判断是抄袭还是重写了一遍。

有些同学，认为结对编程遇到的事件都是相同的，所以写不出来差异。在学术领域和工程领域，单独表述是反抄袭的基本要求。

下面这两位同学结对，报道的是相同的事件，但是措辞、案例、体会都有明显的差异，不单是把二人的名字和你我调换一下而已。

### 例 57

体会：通过这次结对编程，我体会到了不一样的编程体验。一开始我提议使用python，因为我对python比较熟悉。提议使用C++，产生了分歧，后来经过讨论决定使用C++。缩进方面我一直都是Tab党，制定了代码规范，要求4个空格缩进，这次也是狠狠的教训了我一下，在我编写时时刻都要注意自己有没有遵守规范。在对于除零问题上，我开始只认为1~9的运算就能避免出现除零的问题，但实际检验发现9/(1-1)这样也会发生除零的问题，后来决定使用一个判断函数来检测/号后面的数是否为零，为零就重新执行，直到生成的算式没有除零错误。其实编码时人和人的习惯就是不一样，很多地方都不同，这次也谢谢我的结对伙伴同我商讨制定了规范，让我这种平时代码不遵守格式的人学到了很多。

### 例 58

体会收获：通过这次编程，我学习到了结对编程的重要性，以及沟通的重要性。一开始我提议使用C++，提议使用Python，产生了分歧，后来经过讨论决定使用C++。缩进方面主张Tab，我觉得空4个空格比较稳妥，后来经过试验发现Tab确实有时候不稳定，空的长短不一致，就决定按我说的来。除号方面的话，想通过rand()%9+1来避免0，但是我发现这样可能导致所有数取不到0，而且还是后有除0情况发生，比如1/(4-4)，不能完全避免除0，所以我们想了一个判断函数，在计算正确答案的过程中，遇到“/”号就判断除数是否为0，如果是就继续调用生成函数，直到没有除0情况发生。单元测试这些东西也是很新奇的东西，一开始我们没有头绪，摸索花了很长时间，后来逐渐掌握了技巧，就能越来越快的进行测试、修改和再测试。

## 2.13. 预览

下述排版，作者恐怕没有预览过自己的作品。

### 例 59

与，但可以就识别出原单词含义。		
捕捉异常时利用try对用户进行友好的反馈。	W触发3次	规则有效，无更改
大括号应单独占一行，并有相应的大括号在同一列对齐。	W触发1次	强化规则，为增加美观性和代码可读性，大括号对应的if、while和for等也另起一行。
所有的操作符前后加空格。	同时触发多次	规则有效，无更改
变量名、函数名，都只敲一遍，以后全部是拷贝+粘贴。	Z触发3次，经常打错名字造成bug	规则有效，无更改

#### 三、代码规范总结

- 1.第一次进行结对编程。彼此的编程习惯还存在一些差异，需要磨合和统一。
- 2.相关代码的注释可以让彼此更能理解对方的想法。
- 3.变量和方法名的定义要有一定意义，不能随意定义。

## 2.14. 记录完整

### 例 60

每次测试之后要删除无用代码

Z触发2次，W触发3次

规则有效，无更改

虽然是结对编程，但是违反代码规范则一定是每个个体单独完成的。这样的记录才是完整的。

## 2.15. 各种方法

有同学提问过，如果我不会“逆波兰”怎么办？

在具体的问题中，不用逆波兰一样可以解决问题。

例 61

(3).带括号算式的运算，这应该属于这次作业最难的一步了，由于其中有括号的存在，我想到的方法是用递归的思想去求解，就类似于上面添加括号的思想，因为一直是先计算括号内的内容，所以在每次运算只要碰到括号，就递归运算括号里面的，然后在把结果拿出来进行运算，这里有两种方式去运算，我一开始本来是从头到尾一直运算，如果是加减法就继续运算，如果是乘除法就把从这个数计算乘除法，一直到出现加法为止，将得到的数运用第一个数前的符号与一开始得到的书进行运算；另一种

递归可以等价于逆波兰，思路可以。

例 62

功能2：功能2我和张宇讨论的时候想到了用二叉树或者堆栈去存储，然后遇到括号进行匹配。将每个运算符存入其中，如果碰到括号，就等待着与下一个括号进行匹配，然后先对括号里的运算式进行计算。在网络查阅了很多资料，但还是没有实现。

例 63

功能2：应该是采用堆栈的存储方法，将每个元素存入堆栈中，如果碰到括号，就等待着与下一个括号进行匹配，然后先对括号里的运算式进行计算。但是通过查阅资料还是没有实现。

堆栈也可以等价于逆波兰，思想是对的。

还有同学穷举了括号的所有位置，也是个办法。

例 64

难点：如何保证括号成对出现，以及在正确的位置出现（例如不会生成（2）这种情况）。考虑到是四则运算，出现括号的情况较少，我们就采用了枚举的方法。



么解决，答用 MATLAB。题目查重，用 os 标准库的思路也奇怪。这在论文审稿、项目方案选择中，除非有更具说服力的论证，会被否定。

例 69

5.当遇到除0的情况时适用try...except....pass的语句来重新出题。

这种情况，也可以用 if...else 解决，判断分母的值。

比如

例 70

。除号方面的话，通过rand()%9+1来避免0，但是我发现这样可能导致所有数取不到0，而且还是后有除0情况发生，比如01/(4-4)，不能完全避免除0，所以我们想了一个判断函数，在计算正确答案的过程中遇到“/”号就判断除数是否为0，如果是就继续调用生成函数，直到没有除0情况发生。单元测试这些东西也是很新奇的东西，一开始我们没有头绪，摸索花了很久时间，后来逐渐掌握了技巧，就能越

例 71

功能二：

重点：我认为功能二的重点是对于括号的生成

难点：我认为功能二的难点也是对于括号的生成

编程收获：知道如何在编程中如何进行括号的生成

这样的写法，教师/审稿人/项目论证的专业 无法判断你是否能够实现，只有否定。

如果你能在规定时间内学完逆波兰，然后实现，当然很好。如果来不及学习，那么，即使用

1000 万个 IF...ELSE 也不要放弃。对于学习而言，后者更加重要。

例 72

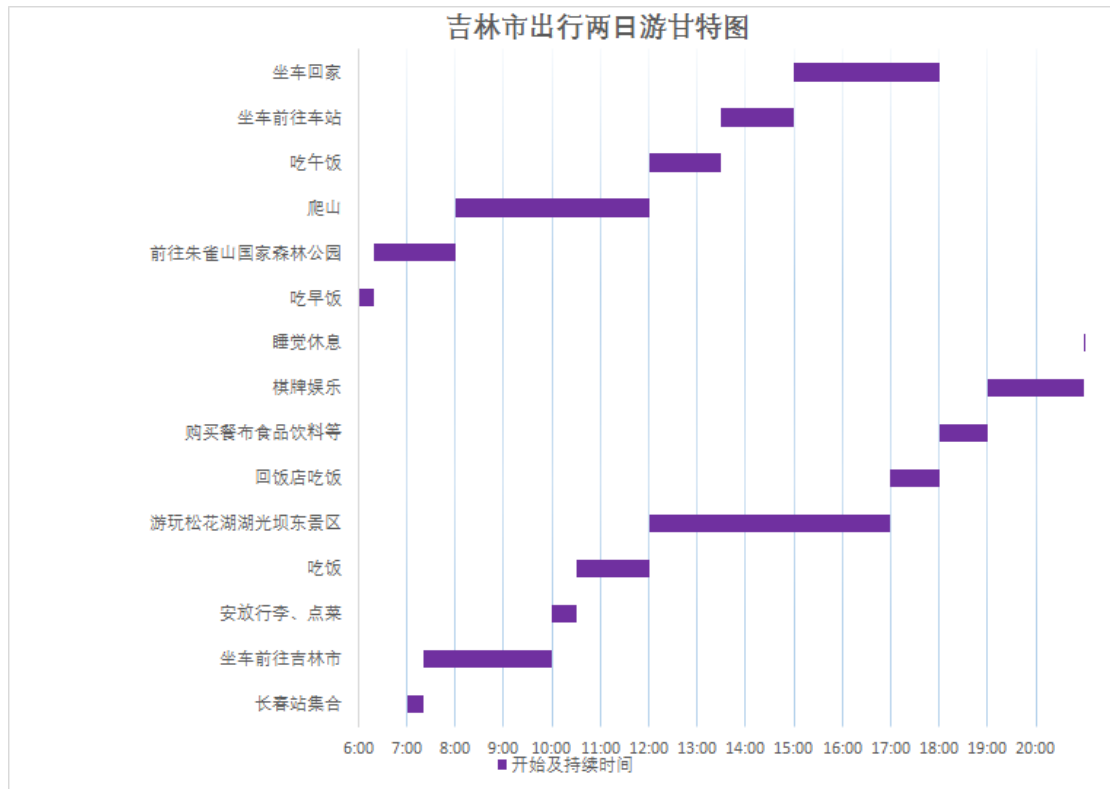
```
elif fh2 == 2:
    if fh3 == 0:
        rjg = n1/n2*n3+n4
    elif fh3 == 1:
        rjg = n1/n2*n3-n4
    elif fh3 == 2:
        rjg = n1/n2*n3*n4
    elif fh3 == 3:
        rjg = n1/n2*n3/n4
elif fh2 == 3:
    if fh3 == 0:
        rjg = n1/n2/n3+n4
    elif fh3 == 1:
        rjg = n1/n2/n3-n4
    elif fh3 == 2:
        rjg = n1/n2/n3*n4
    elif fh3 == 3:
        rjg = n1/n2/n3/n4
```

这样也很好，特别是在问题空间并不很大，可以穷举的情况下。

后来，这两位同学也改用了逆波兰。不过，先做出来再说的态度，非常值得赞赏。如果你有一千个想法，实现不了，也是白扯。有一个想法，即使明知道是笨办法，如果能够解决问题，你就是好的工程师。

## 2.16. 甘特图

例 73

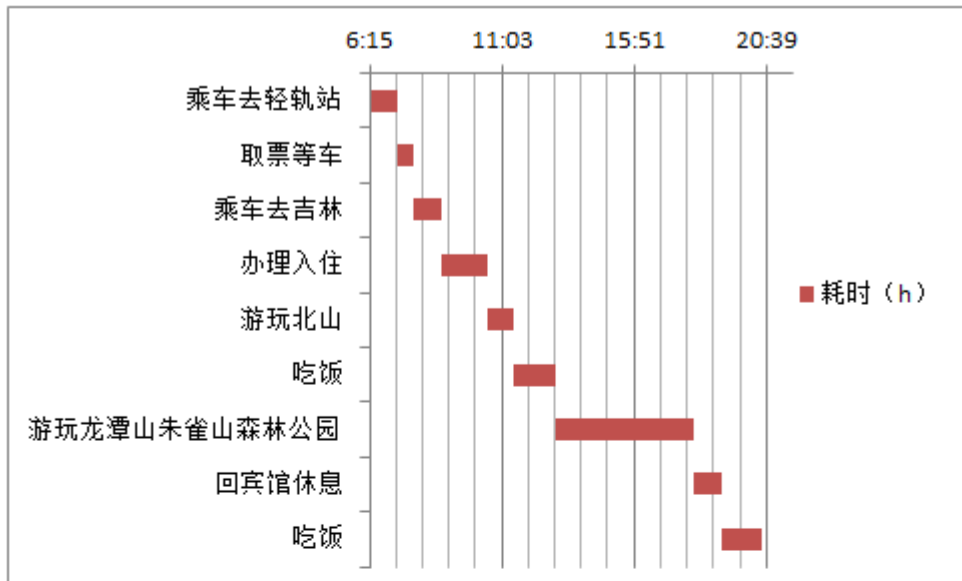


甘特图自上而下，并非按事件年顺序。

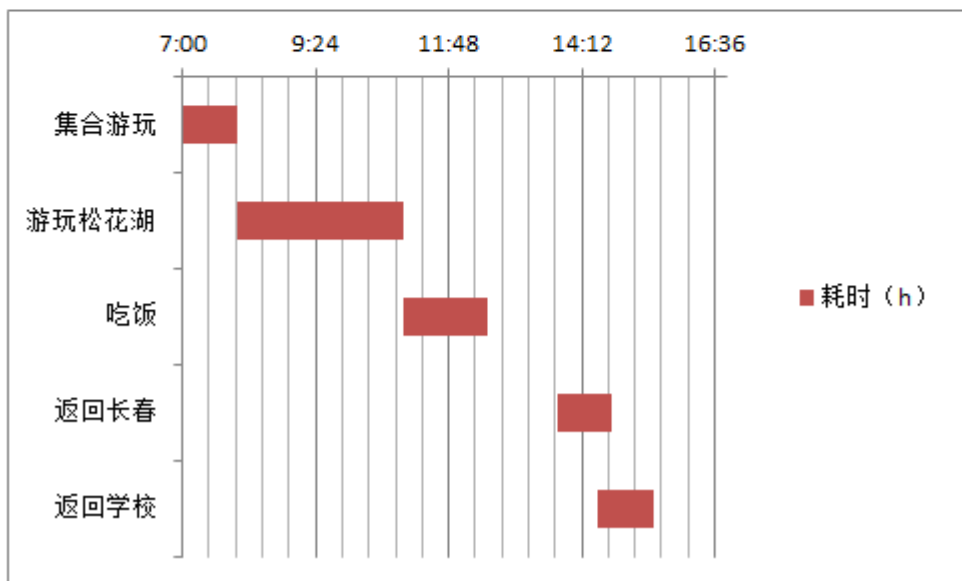
并且，第一天与第二天的在时间上不宜重叠在一张图上。如下图就没有这样的错误暗示。

例 74

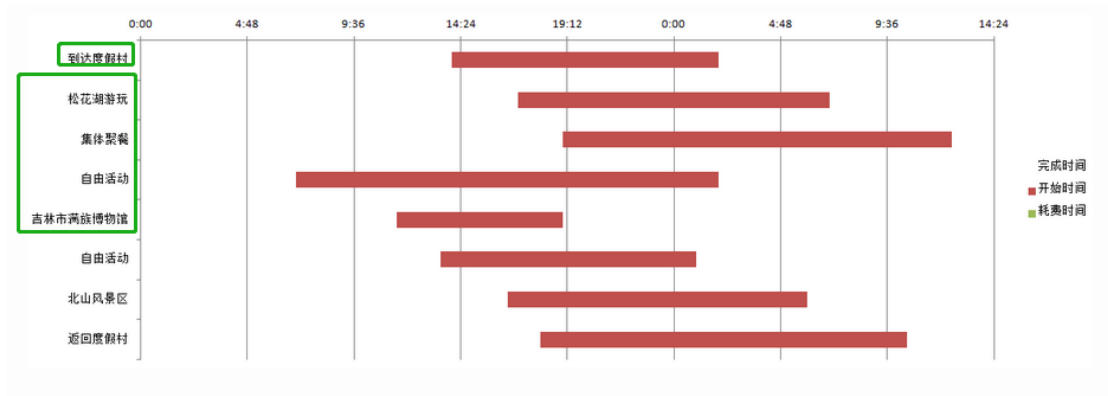
9.30日甘特图



10.1日甘特图

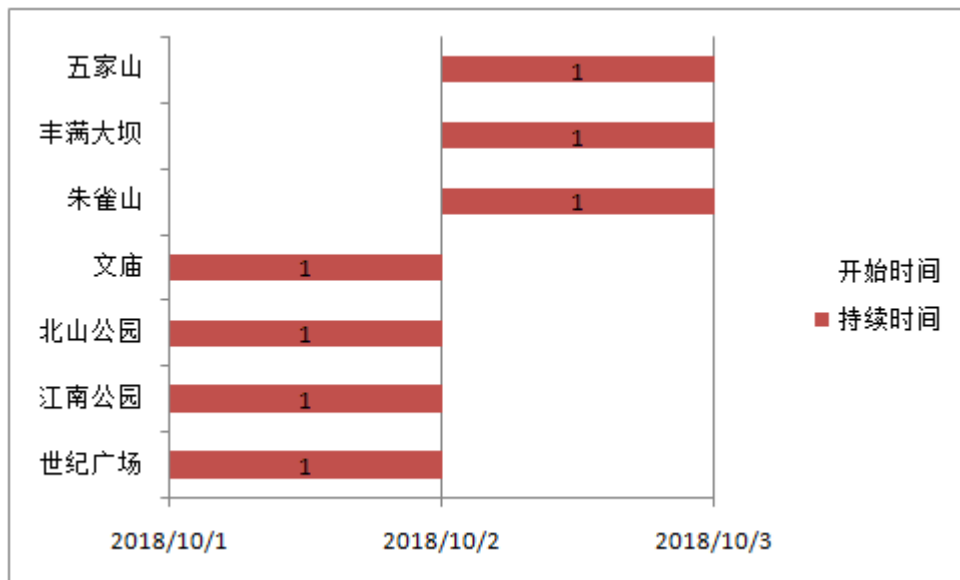






“到达度假村”，不是时间段，而是时间点。若干活动，如绿框圈住的几个，是不能同时安排的。教师怀疑作者没有思考，只是随便做了个甘特图。

例 76



要么，应以小时为分钟为单位细化时间；要么，这里表达的是每个任务都持续一整天。