用Python解决问题

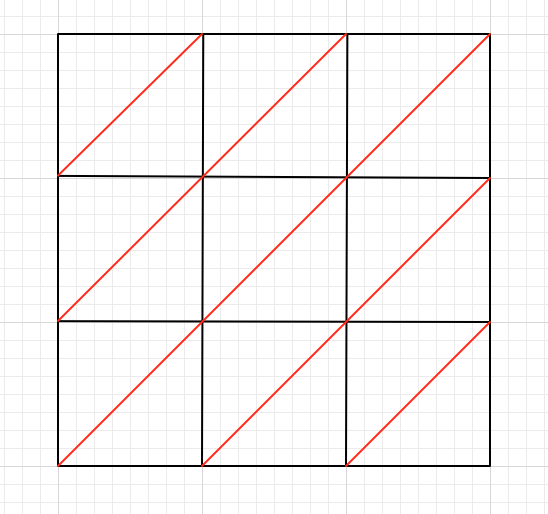
1. 百万皇后问题

尽管课程的要求是要做一些和课程有关的问题，但是大部分专业问题都可以直接利用包去解决，性能分析显得不太有必要。因此我选择了一个较有难度的问题，这个问题虽然不是专业问题，但却是一个十分有趣的问题。对我来说也是一个挑战，因为必须要有一些有效的算法才可以完成。

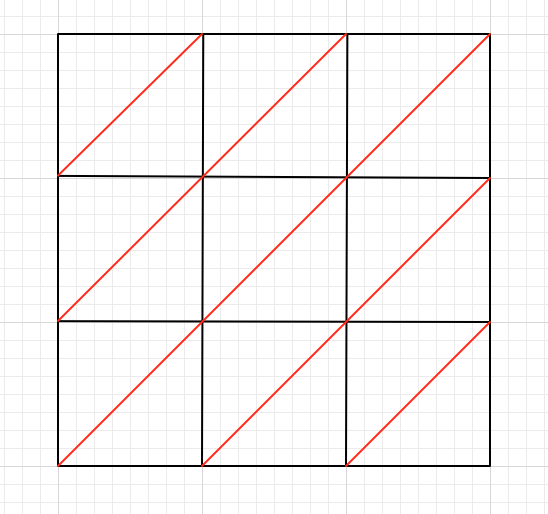
首先介绍一下百万皇后问题，这个问题要做的就是在一个n＊n的棋盘上放置n个皇后，根据国际象棋的规则，这些皇后要彼此之间不能攻击，这就意味着任意两个皇后不可共行，共列或共一斜线。一开始选定这个题目主要是想学习国际象棋，结果搜索到了相关的问题，觉得很有趣，便开始着手尝试解决。

目前n皇后问题的解法有好几种，可以采用随机构造法和枚举法。但是枚举法显然是十分耗时的，百万皇后问题要求的是可行解而非所有的解，因此我们只需要采用随机构造法尽快的找到一个解即可。

然而分析复杂度我发现，在判断一个解是否可行的时候，如果直接扫描判断需要O(n^2)，这样的复杂度显然难以接受。尽管在初始化随机的时候已经保证了行和列均不会有共线的情况，但是这样大的判断复杂度仍然难以完成百万级别的百万皇后问题可行解的求解。因此我才用了三个辅助数组，分别是column，ld，rd。这三者分别是一列，一左横斜行和一右横斜行上皇后的个数。



ld所代表的斜线

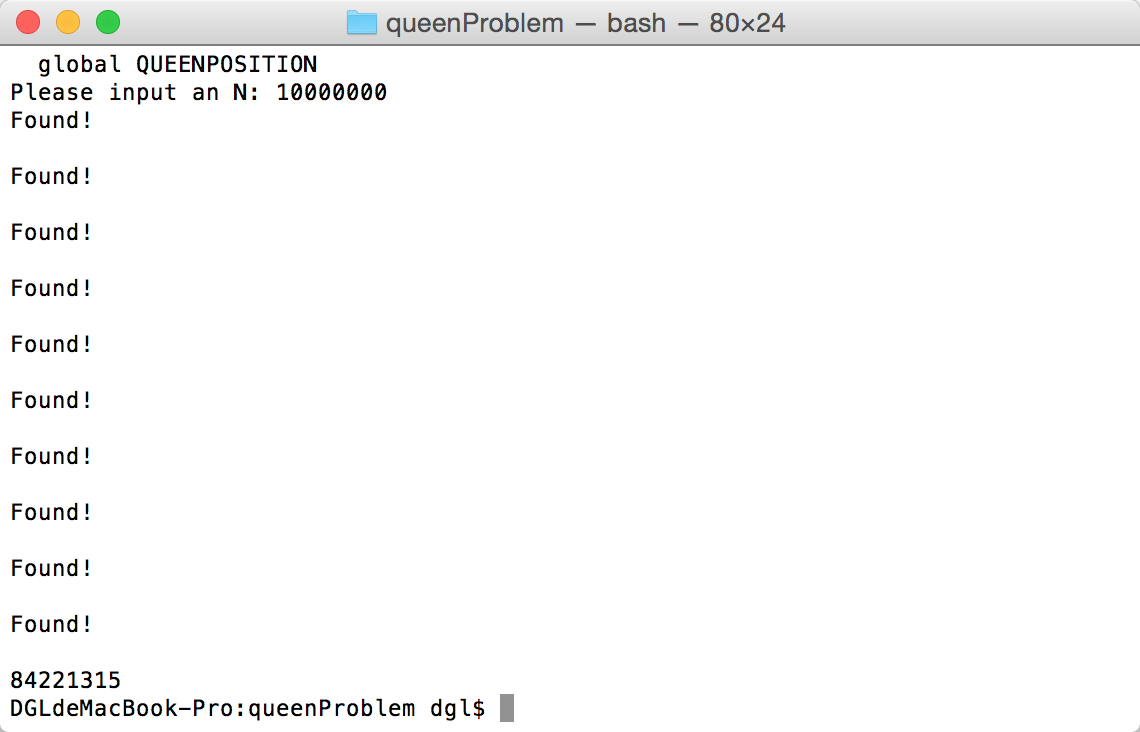


rd所代表的横线

通过这三个数组我们可以做到很好的结果。在程序内部我们利用Python的计数器来计算每次计算的耗时，当然重复十次取总时间的平均值会使结果更具有代表性，毕竟算法有随机的成分在。一百万皇后问题的测试结果如下：



图中显示时间为微秒，也就是说平均耗时大概是8.6秒。然而为了测试我们算法是否优越，也就是算法复杂度是否线性增长，采用千万皇后问题再做一次测试。结果如图：



大概是84.2秒，也就是说这个算法是近似于线性增长的（随着输入规模），这个性能应该说相当好，因此这个任务算是圆满完成了。

1. 热辐射公式的实现与验证

公式背景是实际物体辐射热流量的计算采用斯忒藩-玻耳兹曼定律的经验修正公式 ：

Φ=εAσT4

其中Φ是实际物体辐射热流量

ε 物体的发射率（黑度） 总小于1

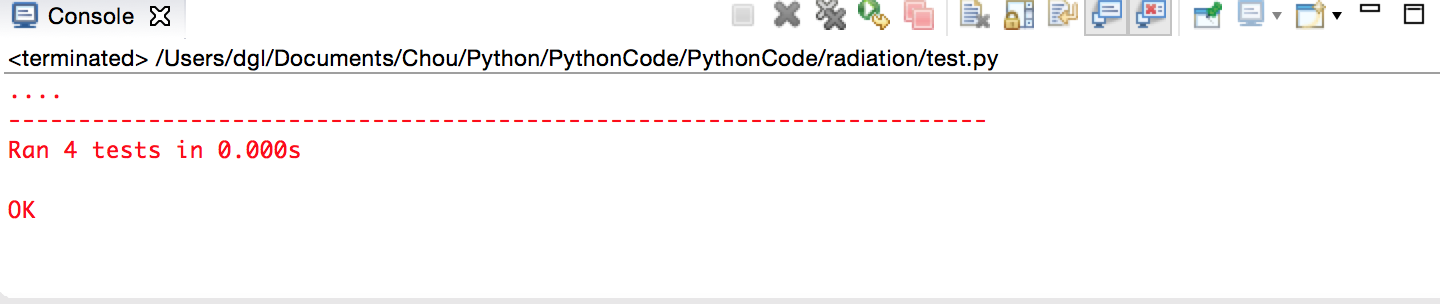
T 实际物体的热力学温度 K

σ 斯忒藩-玻耳兹曼常量，即通常所说的黑体辐射常数，是个自然常数，其值为5.67×10－8（十的负八次方）W/(m2·K4)

A 辐射表面积，m2

在这当中出去一个常量，其余的四个变量可以做到知三求一。我通过python程序验证了这一点，并通过之前学到的unittest对其进行了检验。由于这类数据往往有些许误差，因此本次使用了assertAlmostEqual函数，使得比较的精度有所下降。

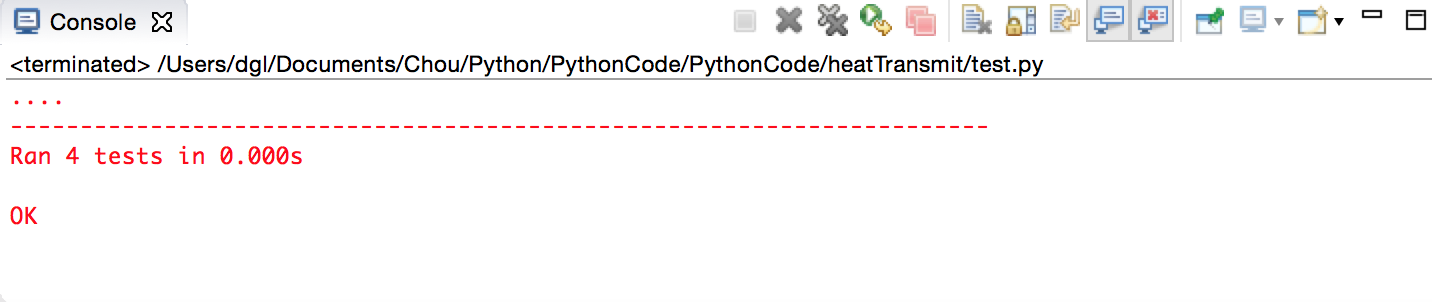
通过radiation中的四个函数，给进三个参数便可以求得剩余的一个，且验证后发现是正确的，结果如下：



1. 热传导公式实现和验证

公式背景是热传导的公式：

Φ=hA, 其中没有常量，根据这个公式，任意给定三个变量可以求得另一个变量的值，我用python实现了，并用unittest测试了一个例子。与上一个基本相似。测试结果如下图：



可见结果是正确的。