

一、实验目标

- 1) 熟悉 C++ 随机数的生成方法[1]，熟悉利用蒙特卡洛方法[2]计算 Pi 值，并熟悉利用随机数生成模拟数据的方法；
- 2) 初步了解程序执行时间的决定因素；
- 3) 比较 vector 和 list 的性能；熟悉容器的操作，包括容器的遍历、插入、删除、排序；熟悉基本的算法函数 sort。

二、实验内容

2.1 第 1 部分：随机数及应用

实验过程：

- 1) 按照 PPT 上的介绍，编写程序计算 Pi 的值。

输入代表包含点数的数字 N，按照 PPT 上的方法计算 Pi 值，并通过加入 stopwatch 记录程序的执行时间。对每个 N 值程序运行 5 次，其中 N 值为 10^4 , 10^5 , 10^6 , 10^7 , 10^8 , 10^9 ，在下表中记录运行的时间（单位秒，结果保留 1 位小数）。最后预测 N 值为 10^{40} 时程序的运行时间。

N 取值	10^4	10^5	10^6	10^7	10^8	10^9
第 1 次						
第 2 次						
第 3 次						
第 4 次						
第 5 次						
平均						
预测	当 $N=10^{40}$ ，预计 Pi 可以精确到____位，执行时间为____秒。					
机器配置	CPU: 内存: 操作系统:					

特别提示:①浮点计算是非常耗时的操作，求平方根操作更是如此。要使计算时间最短，需要优化程序——在保证正确的前提下，有效降低计算复杂度；②随机数的生成需要做随机化操作（见 PPT 第 77-78 页，没有介绍，需要自学），否则表格中 Pi 值估计对同样的 N 值都是一样的；③特别注意 int 型的数据表示范围，以及 double/real 类型的数据精度范围。

2.1-1 提交要求：填写表格，并提交源代码 E1_1.cpp。

*2) 选择一种你熟悉的脚本语言（例如 Python, Perl, PHP, 等），按照 1) 中的要求重复实验，填写下表，并于上面的结果进行比较（不同的脚本语言统计执行时间不一样，请自行查阅网络资料）。

N 取值	10^4	10^5	10^6	10^7	10^8	10^9
第 1 次						
第 2 次						
第 3 次						
第 4 次						
第 5 次						
平均						
预测	当 $N=10^{40}$ ，预计 Pi 可以精确到____位，执行时间为____秒。					

机器配置	CPU:	内存:	操作系统:
------	------	-----	-------

2.1-2 提交要求: 填写表格, 并提交源代码 E1_2.py/pl 等

2.2 第2部分: 学生成绩分析。

实验过程:

1) 生成一个姓名文件。

姓名文件的要求: 姓名总共 20000 个, 而且满足: ①名字选择: 从 wordlist.txt (wordlist 来自 [3], 做了适当的处理, 共有约 10W 单词) 选取; ②具体规则如下: 去掉所有字符数个数少于 10 个的单词; 去掉所有字符个数多余 12 个的单词; 余下的单词按照字典序排序, 选择前 20000 个; ③把程序的输出定向到 20knames.txt 文件。

2.2-1 提交要求: 提交源代码 E2_1.cpp 及相关文件。

2) 生成学生成绩列表。

要求学生有 20000 名 (来自 1) 中的输出结果), 每个学生的成绩如下 (黑体行不在其中):

```
#name      期中 期末 平时成绩 (6 次)
abandonedly 85   84   70 89 75  81  94  94
```

每个学生成绩占一行。其中: ①名字的选择: 从 20knames.txt 读取, 通过输入重定向的方式完成; ②期中的成绩均匀分布在 [60, 90], 期末的成绩均匀分布在 [50, 85], 平时成绩均匀分布在 [70, 95]; ③把程序的输出通过重定向的方式写入文件 20kscores.txt。

2.2-2 提交要求: 提交源代码 E2_2.cpp 及相关文件。

3) 读入上述生成的 20kscores.txt 文件, 按照如下方式计算每个学生的成绩

总成绩 = 0.4 * 期中 + 0.4 * 期末 + 0.2 * 平时成绩的中值

并把学生按照总成绩从高到低排列, 按照如下格式通过 cout 输出每个学生的成绩 (黑体行不在其中, 且总成绩保留 1 位小数)。

```
#name      总成绩 期中 期末 平时成绩 (6 次)
meticulous  87.6   89   83   94   94  91  80  94  94
```

在上述处理中, 分别使用 vector 和 list, 并对从输入、计算、输出的全过程通过 stopwatch 统计时间; 同样的统计应该至少进行 5 次并计算平均值作为最后比较的标准。统计的时间数据通过 clog 输出。

2.2-3 提交要求: 提交源代码 E2_3.cpp 及相关文件, 并提交结论: 使用 vector 还是 list 更佳。

2.3 思考题

按照上述要求完成各部分的程序。如果 E221.exe、E222.exe 和 E223.exe 分别代表 2.2 中 1)、2) 和 3) 生成的执行程序, 把这些程序放在同一目录下, 并把 wordlist.txt 也放在该目录下, 在控制台下运行下面的命令并观察输出结果

```
E221 < wordlist.txt | E222 | E223
```

上面的|称为管道, 会把左侧程序的输出变成右侧程序的输入, 上面的命令构成了一个线性的输入->处理->处理->...->输出的链条。

参考资料

- [1] C++随机数。 <https://blog.csdn.net/luotuo44/article/details/33690179>
- [2] 蒙特卡罗方法。 <http://www.ruanyifeng.com/blog/2015/07/monte-carlo-method.html>
- [3] 英文单词表。 <http://www-01.sil.org/linguistics/wordlists/english/>

2020 年 OOP 上机实验(3)

提交检查表

2.1 提交要求

2.1-1 提交要求：填写表格，并提交源代码 E1_1.cpp。

N 取值	10^4	10^5	10^6	10^7	10^8	10^9
第 1 次	1.0×10^{-3}	5.0×10^{-3}	5.2×10^{-2}	5.2×10^{-1}	5.4	53.9
第 2 次	1.0×10^{-3}	7.0×10^{-3}	6.1×10^{-2}	6.0×10^{-1}	5.4	54.0
第 3 次	1.0×10^{-3}	6.0×10^{-3}	6.7×10^{-2}	5.3×10^{-1}	5.4	53.8
第 4 次	1.0×10^{-3}	7.0×10^{-3}	5.0×10^{-2}	5.3×10^{-1}	5.3	54.5
第 5 次	1.0×10^{-3}	5.0×10^{-3}	5.4×10^{-2}	5.2×10^{-1}	5.3	53.7
平均	1.0×10^{-3}	6.0×10^{-3}	5.68×10^{-2}	5.4×10^{-1}	5.36	53.98
预测	当 $N=10^{40}$ ，预计 Pi 可以精确到 <u>8</u> 位，执行时间 5.4×10^{32} 秒。					
机器配置	CPU: i5 内存: 8GB 操作系统: win10 (64 位)					

2.1-2 提交要求：填写表格，并提交源代码 E2_2.py/pl 等

N 取值	10^4	10^5	10^6	10^7	10^8	10^9
第 1 次	5.6×10^{-3}	7.1×10^{-2}	6.4×10^{-1}	6.4	61.3	621.6
第 2 次	5.5×10^{-3}	7.7×10^{-2}	6.5×10^{-1}	6.2	61.0	620.4
第 3 次	5.7×10^{-3}	5.6×10^{-2}	6.4×10^{-1}	6.0	60.8	622.0
第 4 次	5.3×10^{-3}	7.6×10^{-2}	6.4×10^{-1}	6.3	61.2	622.4
第 5 次	5.3×10^{-3}	6.0×10^{-2}	6.4×10^{-1}	6.4	61.4	621.8
平均	5.48×10^{-3}	6.8×10^{-2}	6.42×10^{-1}	6.26	61.14	621.64
预测	当 $N=10^{40}$ ，预计 Pi 可以精确到 <u>15</u> 位，执行时间为 6.2×10^{33} 秒。					
机器配置	CPU: i5 内存: 8GB 操作系统: win10 (64 位)					

2.2 提交要求

2.2-1 提交要求：提交源代码 E2_1.cpp 及相关文件。

2.2-2 提交要求：提交源代码 E2_2.cpp 及相关文件。

2.2-3 提交要求：提交源代码 E2_3.cpp 及相关文件，并提交结论：使用 vector 还是 list 更佳。(vector 更佳，因为用到的操作主要是顺序访问和索引，没有使用插入和删除等操作)