

成绩	
----	--



北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

本科生《编译原理》课程实践报告

题 目： 程序设计语言认知实验

学 院： 徐特立学院

专业名称： 计算机科学与技术

姓 名： 陈照欣-1120191086

目录

实验目的.....3

实验内容.....3

环境配置.....3

实验的具体过程和步骤.....3

实验截图.....3

实验结果.....5

语言易用性和程序规模对比分析.....6

实验心得体会6

实验目的

了解程序设计语言的发展历史，了解不同程序设计语言的各自特点；感受编译执行和解释执行两种不同的执行方式，初步体验语言对编译器设计的影响，为后续编译程序的设计和开发奠定良好的基础。

实验内容

对于矩阵乘法这一功能，具体要求为：对于给定的矩阵规模，自动生成两个随机整数矩阵，计算两矩阵的乘积，输出结果。分别使用 C/C++、Java、Python 和 Haskell 实现，对这几种语言实现的编程效率、程序规模、程序的运行效率进行对比分析。

环境配置

处理器 Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz
机带 RAM 12.0 GB (11.8 GB 可用)
设备 ID FDE9AABD-B6A6-4EB0-9805-DE4193E9D97B
产品 ID 00331-10000-00001-AA098
系统类型 64 位操作系统, 基于 x64 的处理器
笔和触控 没有可用于此显示器的笔或触控输入
gcc (tdm64-1) 9.2.0
python 3.8.8
openjdk 11.0.14.1 2022-02-08
OpenJDK Runtime Environment Temurin-11.0.14.1+1 (build 11.0.14.1+1)
OpenJDK 64-Bit Server VM Temurin-11.0.14.1+1 (build 11.0.14.1+1, mixed mode)
GHC 8.8.4
Visual Studio Code 1.65.0

实验的具体过程和步骤

Step 1: 设定矩阵的规模分别为 200×300 和 300×400 ，每个元素的范围为 $1 \sim 100$ ；
Step 2: 规定程序运行时间为输入矩阵规模后到结果输出完成之间的时间间隔；
Step 3: 对于每一个程序，执行 5 次，记录运行时间以及程序占用的内存大小求平均值

实验截图

```
experient1.cpp x
experient1.cpp > main()
44     }
45     for(int i=0;i<M1;i++)
46     {
47         for(int j=0;j<M2;j++)
48         {
49             int sum=0;
50             for(int k=0;k<M1;k++)
51                 sum+=matrix1[i][k]*matrix2[k][j];
52             temp.push_back(sum);
53         }
54         ans.push_back(temp);
55     }
56     DWORD End = GetTickCount();
57     for(int i=0;i<M1;i++)
58     {
59         for(int j=0;j<M2;j++)
60             cout<<ans[i][j]<<" ";
61         cout<<endl;
62     }
63
64     cout << "The run time is: " <<End - Start << "ms" << endl;
```

问题 ① 输出 调试控制台 终端

```
273 742425 706976 669587 692630 721652 674110 698092 742715 707815 745959 694752 700046 686238 720297 732714 719012 651178 664017 742701 68932
7 740516 697379 695067 743898 683292 715054 717461 727675 654266 716140 689453 750498 782487 687429 761836 704358 738493 665654 696771 705850
757914 748411 711868 731273 759510 732804 757036 709544 713640 727857 734387 728186 702237 714765 745074 707282 715395 737732 705837 692935 70
4527 734366 745215 721490 738788 698580 733114 743577 681481 731348 678704 719648 759876 679982 734491 695395 712630 714137 727248 726836 7402
23 749244 73427 707610 745226 732308 725610 716692 768891 687397 683675 662139 725399 713599 721877 726515 723162 721527 710758 713063 679662
724426 719928 715719 748067 746847 725674 730428 753834 697532 710207 693221 720305 728976 719499 741406 683140 712836 731140 734104 744495 7
22135 704850 660635 677574 685574 758822 696258 700405 719767 704739 717070 683650 742678 701129 715561 699325 716174 699585 709327 739182 733
800 654265 731010 742725 734961 735282 730902 692644 676356 732280 704106 718034 737350 744327 716429 726341 701505 684263 723874 735653 70719
4 754205 748854 782403 672595 712420 719300 680584 711086 747939 686065 666550 706606 716014 683951 707126 676676 709380 682013 686045 770647
726353 749784 717434 743762 719058 716617 690570 727502 744851 695814 717395 727402 700642 712389 749917 695929 732463 730051 758106 738089 68
1296 713767 731121 714980 714247 695747 694331 684395 710902 764574 729856 723518 753753 757981 710526 701200 703256 691823 749077 708451 7476
17 673427 660103 752010 712604 691042 769067 706867 739517 717047 711022 718129 704510 765041 723602 710853 685070 719686 719432 747779 683324
727475 704334 692992 689177 697271 720695 782038 678458 689400 797064 713452 734635 690894 739963 706219 694729 691957 682883 679423 751184 6
93937 722619 726569 724927 692973 648122 746975 693201 738475 696930 700141 703850 792719 706875 689695 710955 723364 716442 719801 731055 712
779 710575 730947 699000 690538 734522 704230 711840 730728 713293 682745 729008 739572 703106 742992
The run time is: 187ms
```

```
experient1.py x
experient1.py ...
4 matrix1, matrix2 = [], []
5 N1, M1, N2, M2 = map(int, input("Please input the size of matrix:").split())
6 while (M1 != M2):
7     print("Invalid Input! Please input again:")
8     N1, M1, N2, M2 = map(int, input().split())
9
10 start = time.time()
11 nums=[]
12 for i in range(N1):
13     for j in range(M1):
14         nums.append(random.randrange(100))
15     matrix1.append(nums)
16 nums.clear()
17 for i in range(N2):
18     for j in range(M2):
19         nums.append(random.randrange(100))
20     matrix2.append(nums)
21 # print(matrix1)
22 # print(matrix2)
23
24 ans = []
```

问题 ② 输出 调试控制台 终端

```
7668, 1298552, 285048, 1092340, 1488584, 823472, 1472748, 174196, 855144, 791800, 1203536, 411736, 1235208, 199032, 0, 332556, 744292, 886816,
142524, 316720, 1314388, 15836, 807636, 902652, 585932, 1520256, 1551928, 1108520, 174196, 712630, 174196, 886816, 95016, 902652, 855144, 300
884, 902652, 253376, 1488584, 839308, 1108520, 1441076, 31672, 237540, 31672, 1377732, 585932, 79180, 1441076, 570096, 475080, 475080, 427572,
855144, 332556, 269212, 807636, 1472748, 902652, 981832, 79180, 981832, 934324, 712620, 380064, 1187700, 1536092, 728456, 665112, 110852, 158
36, 886816, 712620, 95016, 649276, 1282716, 1504420, 997668, 712620, 965996, 1377732, 1298552, 142524, 585932, 459244, 1251044, 1235208, 15836
, 126688, 839308, 411736, 601768, 886816, 981832, 1251044, 364228, 1377732, 1203536, 411736, 205868, 1314388, 443408, 443408, 1536092, 1456912
, 1504420, 1551928, 886816, 807636, 475080, 475080, 1456912, 1298552, 775964, 680948, 31672, 934324, 1567764, 0, 1013504, 1266880, 522588, 0
, 1092684, 174196, 1488584, 1472748, 1567764, 79180, 522588, 934324, 601768, 855144, 253376, 47508, 269212, 1536092, 63344, 1456912, 1330224,
1235208, 1504420, 870980, 300884, 1187700, 1488584, 490916, 696784, 126688, 1156028, 997668, 253376, 1314388, 348392, 269212, 1076848, 839308
, 1203536, 997668, 665112, 237540, 427572, 79180, 1061012, 839308, 1536092, 522588, 839308, 649276, 1251044, 1330224, 570096, 1203536, 633440
, 95016, 110852, 1156028, 934324, 870980, 712620, 1029340, 522588, 1456912, 823472, 300884, 1171864, 443408, 522588, 1013504, 316720, 300884, 2
53376, 1108520, 364228, 364228, 443408, 1266880, 316720, 1488584, 506752, 807636, 1061012, 1298552, 823472, 601768, 63344, [1171864, 1156028,
1425240, 1425240, 1330224, 807636, 617604, 1156028, 1251044, 1013504, 443408, 1266880, 1456912, 839308, 1061012, 902652, 981832, 855144, 5542
60, 1361896, 1551928, 665112, 1346060, 934324, 380064, 1013504, 1251044, 981832, 1393568, 1567764, 411736, 459244, 63344, 1456912, 443408, 395
980, 696784, 839308, 1346060, 680948, 601768, 459244, 63344, 696784, 965996, 997668, 918488, 31672, 1456912, 3
time cost: 10.538080275360107 s
PS D:\Documents\Compilers Principles\experient1>
```

powershell C/C++: ... cpplibg: ex... Java Debug ... Python Deb...

行 14, 列 24 空格: 4 UTF-8 CRLF Python 3.8.8 (base: conda)

experient1.java 1 X

experient1.java > experient1 > main(String[] args)

```
27  a[i][j]=rand.nextInt(100);
28
29  for(int i=0;i<M2;i++)
30  for(int j=0;j<M2;j++)
31  b[i][j]=rand.nextInt(100);
32  for(int i = 0;i<a.length;i++)
33  {
34      for(int j = 0;j<b[0].length;j++)
35      {
36          for(int k = 0;k<a[0].length;k++)
37          {
38              c[i][j] += a[i][k]*b[k][j];
39          }
40      }
41  }
42  // System.out.println("Matrix a: ");
43  // for(int i = 0;i<a.length;i++)
44  // {
45  //     for(int j = 0;j<a[0].length;j++)
46  //     {
47  //         System.out.printf("%d",a[i][j]);
48  //     }
49  // }
```

问题 ① 输出 调试控制台 终端

583, 746518, 748532, 797152, 709434, 688926, 728697, 651454, 709898, 733582, 741055, 699562, 727836, 759562, 702800, 683519, 695449, 696394, 710609, 719255, 68046
9, 685043, 739435, 696015, 722335, 751736, 729204, 769897, 604768, 711240, 761630, 758621, 690065, 708368, 748558, 805195, 744798, 746751, 686435, 714467, 705048,
717540, 711912, 751364, 698877, 763300, 725501, 776641, 716485, 706969, 746571, 711015, 721906, 723665, 708005, 755492, 683206, 729468, 771402, 741046, 679006, 68
9405, 721186, 696536, 676300, 713412, 706288, 661640, 749775, 713221, 731835, 714554, 685422, 734003, 693093, 723315, 662567, 757270, 736025, 741855, 708480, 6956
76, 698975, 703605, 672643, 706401, 732790, 712471, 710651, 672399, 706633, 727502, 708321, 674310, 700998, 740631, 699892, 686803, 704869, 715278, 719936, 721856
, 751709, 734612, 703244, 715187, 733609, 742379, 780127, 697108, 701624, 671803, 735988, 685117, 728218, 745651, 697205, 701568, 699146, 698204, 744705, 707382, 6
94566, 688087, 747414, 749682, 721807, 764022, 696335, 740187, 684173, 710812, 724169, 723842, 676098, 725254, 745078, 716357, 728056, 716724, 666970, 724790, 674
037, 684265, 742103, 642780, 740314, 689169, 713686, 717717, 735919, 710976, 717262, 699024, 728170, 690437, 692693, 723251, 718130, 779303, 727994, 676398, 69832
6, 726267, 716698, 698032, 713168, 742185, 739481, 732562, 744864, 701545, 737947, 760609, 687318, 721744, 705187, 741972, 706624, 768119, 690653, 736462, 699692,
674670, 692646, 702551, 692741, 735289, 755654, 703575, 760874, 733167, 662035, 731872, 656625, 705768, 726927, 687265, 767384, 687951, 692596, 681839, 732214, 77
8242, 674977, 678432, 745246, 711053, 708528, 676578, 676517, 675620, 746718, 745908, 778658, 709270, 709042, 692893, 713639, 766034, 745363, 700433, 713581, 6778
73, 750177, 690708, 707439, 753021, 704784, 756596, 740088, 693194, 682773, 678047, 688590, 678693, 704854, 724115, 770142, 709505, 753745, 726242, 755995, 690452
, 689714, 716373, 690978, 673704, 760815, 710525, 693176, 688599, 712755, 701307, 688937, 743383, 669414, 734044, 699105, 719620, 741404, 706635, 685307, 681305, 7
30027, 722333, 753621, 668046, 728828, 705115, 701653, 705557, 669417, 669350, 702502, 652488, 714929, 739772, 684233, 777823, 748428, 701802, 741746, 724295, 711
365, 743851, 712879, 704263, 707814, 700004, 715119, 740397, 692463, 738423, 652843, 729862, 661537, 678288, 676492
Running Time: 50ms

Lib.hs X Lab 1. 程序设计语言认知实验.pdf

experient1 > src > Lib.hs > main

```
40
41  main :: IO ()
42  main = do
43      s <- getCPUtime
44      matrix1<-randmul1 200
45      let matrixa = init matrix1
46      matrix2<-randmul2 300
47      let matrixb = init matrix2
48      --print $ matrixa
49      --print $ matrixb
50      mapM_ print $
51      | multiply matrixa matrixb
52      e <- getCPUtime
53      putStrLn $ show (fromIntegral (e - s) / 10^12) ++ " seconds"
```

问题 ② 输出 调试控制台 终端

085627, 3023287, 2889865, 3134527, 3061958, 2984046, 2981864, 2881601, 2986497, 2908273, 2816984, 2838024, 2903738, 3026551, 2745680, 2711388, 2846046, 2683253
, 2857304, 2655540, 2805622, 2736910, 2805676, 2839020, 2603403, 2767845, 2867130, 2755061, 2918415, 2563819, 2845640, 2825465, 2901631, 2480950, 2713890, 25804
67, 2595443, 2630028, 2595605, 2411472, 2795875, 2716212, 2620097, 2633030, 2656491, 2354427, 2448911, 2566101, 2509219, 2541400, 2412094, 2575930, 2567528, 243
1672, 2356886, 2530779, 2395015, 2419643, 2581251, 2367591, 2364909, 2510287, 2346256, 2316329, 2335044, 2547920, 2365687, 2394489, 2170795, 2324545, 2247969, 2
225694, 2121388, 2055782, 2243452, 2126297, 2255635, 2134049, 2118588, 2031873, 2053787, 2069114, 2137313, 2281745, 2050507, 2118898, 2188644, 2122783, 2284707
, 2126262, 1959475, 2091264, 2067526, 2054462, 1987133, 2062403, 1876706, 2036567, 1912650, 2036435, 1981205, 1881375, 1955861, 1837956, 1884608, 1873990, 18891
55, 1867935, 1961188, 1878487, 1808776, 1723668, 1873118, 1740334, 1835822, 1812097, 1898524, 1765440, 1887831, 1660538, 1727079, 1772213, 1794878, 1609228, 160
8724, 1587483, 1620600, 1549079, 1604662, 1730507, 1705750, 1571741, 1732140, 1672836, 1577196, 1517250, 1430936, 1486560, 1511569, 1448190, 1605356, 1497936, 1
478986, 1371130, 1539537, 1534265, 1378029, 1493672, 1372886, 1426092, 1451452, 1408118, 1365734, 1474071, 1382425, 1349064, 1237125, 1298111, 1269719, 1330781
, 1309135, 1191287, 1209667, 1268886, 1158933, 1269606, 1149103, 1273912, 1121315, 1177200, 1256750, 1211957, 1131076, 1162529, 1082088, 1030561, 1172108, 10538
87, 1043079, 1135391, 1159621, 1114221, 1098521, 1010815, 1063105, 1036739, 925765, 904132, 1006129, 968384, 977496, 984519, 937875, 916236, 907450, 899564, 8686
72, 910302, 927307, 876739, 826942, 829234, 841447, 807608, 835943, 853442, 789565, 789771, 765967, 745212, 716068, 717088, 727954, 687968, 677440, 692377, 636565
, 681629, 661387, 670689, 654521, 667291, 592258, 589432, 543729, 541925, 500610, 577262, 512714, 538174, 533416, 480162, 460377, 461155, 455134, 490063, 452806, 4
66989, 445261, 433475, 386281, 414750, 388855, 379109, 388843, 372690, 348497, 336936, 334077, 307893, 294500, 285307, 284756, 264230, 255710, 237379, 227251, 224
870, 213737, 206770, 195472, 178282, 155994, 153345, 143024, 130730, 120379, 111879, 94115, 86762, 75553, 65285, 58067, 42578, 32745, 21970]
10. 28125 seconds
*Main>

实验结果

C++:

	第一次	第二次	第三次	第四次	第五次	平均
--	-----	-----	-----	-----	-----	----

运行时间	187ms	170ms	183ms	200ms	179ms	186ms
占用内存	32.5MB	34MB	32.5MB	31MB	25MB	31MB

Python

	第一次	第二次	第三次	第四次	第五次	平均
运行时间	6.15s	5.48s	5.52s	5.75s	6.04s	5.79s
占用内存	28MB	25MB	18MB	23MB	30MB	25MB

Java

	第一次	第二次	第三次	第四次	第五次	平均
运行时间	50ms	45ms	46ms	45ms	54ms	48ms
占用内存	80MB	76MB	79MB	77MB	83MB	79MB

Haskell

	第一次	第二次	第三次	第四次	第五次	平均
运行时间	10.32s	9.98s	10.20s	10.48s	9.11s	10.01s
占用内存	690.97MB	670.88MB	701.32MB	705MB	669.17MB	687.47MB

语言易用性对比分析

- (1) C 语言是一个通用的命令式计算机程序设计语言，属于静态强类型，支持结构化程序设计，是操作系统、数据库和嵌入式等系统级软件开发经常使用的编程语言。C 语言能以简易的方式编译、处理低级存储器。C 语言是仅产生少量的机器语言以及不需要任何运行环境支持便能运行的高效程序设计语言。尽管 C 语言提供了许多低级处理的功能，但仍然保持着跨平台的特性，以一个标准规格写出的 C 语言程序可在包括类似嵌入式处理器以及超级计算机等作业平台的许多计算机平台上进行编译。C 语言是一个命令式面向过程的编程语言，对应的高级语言程序很容易被翻译到机器语言执行，相对其他语言执行效率较高。
- (2) 由于 C++ 更接近底层的特性，C++ 的语法较 Java 更加复杂。Java 和 C++ 虽然都有类的概念，但是 Java 的功能更加丰富，且不受平台限制。Java 继承了 C 和 C++ 的很多语法，但是抛弃了较多的低层次的内容，例如指针和内存操作等。Java 中提供了垃圾收集器，编写程序时不需要再手动删除创建的对象。
- (3) Python 有丰富的库可以调用，且支持算法测试，开发速度更快，适用于后端开发、数据科学、机器学习等领域。但是运行时间相对较长，且对解释器的要求较高。
- (4) Haskell 是静态强类型系统，是一个标准化的纯函数式编程语言。主流实现 GHC 既支持解释执行又支持翻译执行，因此效率比 python 略高。作为带类型的函数语言，适用于数据处理，符号计算，编译器。但是由于其独特的语法和国内较小的使用范围，Haskell 对于新手不太友好。

程序规模对比分析

python, c++ 实现的规模近似，都需要数据的定义，三层 for 循环实现结果，规模大致在 15-20 行。在代码量上 haskell 的优势非常明显。

实验心得体会

编程语言的选择通常依赖于开发速度、执行速度、鲁棒性、受众面等多个因素。不同领域对这些因素的要求不同，适合的语言也不同。对于操作系统、网络、硬件驱动等接近底层的开发，靠近硬件的 C 比较适合。金融电商公司等对于代码稳定性的要求较高，易于跨平台和代码优化的 Java 是更好的选择。此外，由于 C/C++ 和 Java 的对编译环境的要求较低，编译器较为简洁，且语法通俗易懂，故更适合新手学习。在科研领域，需要各种强大的工具包，那么可以选择 python。Haskell 的复杂是其能很好的控制大型程序，在设计大型程序时，用 Haskell 语言能较容易的把握核心复杂度。Haskell 的复杂实际上是大型程序本身的复杂度在程序语句上的直接体现。

附件

Lab1_haskell

Lab1_java

Lab1_C++

lab1.py