# 实验报告

## 一、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。

学习和使用C/C++、Python、Java、Haskell四种编程语言，并用四种语言编写实现同一功能的简单程序，并体会不同的编程语言的语言、语法特点，了解不同编程语言的特性、运行效率的高低、代码规模大小等特征。通过实际编程实验，横向对比不同语言之间的特点和特性。

## 二、实验内容

分别使用 C/C++、Java、Python 和 Haskell实现快速排序。对采用这几种语言实现的变成效率、编程规模、程序运行效率进行对比分析。

## 三、实验步骤

### 1、环境搭建

本实验在Ubuntu 18.04.4 LTS系统下进行。apt-get命令提供了Linux系统下，对软件的管理，可以方便的完成各个语言环境的安装。通过使用相应的命令，使用Linux系统的程序管理器，即可快速完成对实验环境的搭建。使用Visual Studio code作为实验代码的编辑器。使用的各语言运行环境版本如下：

* C/C++：gcc 7.5.0
* Java：openjdk 1.9.0\_282
* Python：Python 3.6.9
* Haskell：GHCI 8.0.2v

### 2、数据准备

预先使用Python语言编写简单程序，生成测试数据。使用random库，生成一个由500000个0～10000间数字组成的随机数组作为测试数据，并保存为：5000000.txt。同时，生成一个大小为7的数据用于验证程序正确性，保存为：7.txt。

### 3、代码编写

使用四种编程语言，对快速排序进行实现。快速排序是一种时间复杂度为O(logn)的快速排序算法。它使用递归的方式进行实现，可以快速完成对数据的排序。为了方便测试，将待排序的数据存储在.txt文件中，使用文件操作读入这些数据并完成排序。使用不同语言所提供的时间模块完成对排序过程的计时。

对于四种语言，均使用函数的方式封装快速排序算法为qsort(),在主程序中完成对数据文件的读取、排序计时等操作。

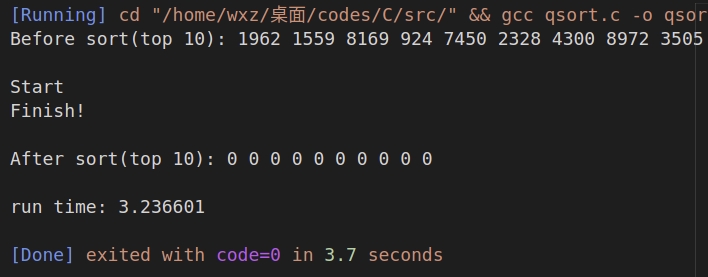
在代码开发效率方面，Python语言使用最为简单，开发效率最高。C、Java语言的开发效率相近。由于它们的语法要求较为严格，因此需要耗费一定的时间进行调试。Haskell语言的语法较为特殊，因此在涉及到循环、函数操作时需要花费更多的时间进行设计，开发效率偏低。

### 4、实验操作

Visual Studio code中的Code Runner插件提供了对上述语言运行的支持。使用此插件实现代码的执行。分别执行四个程序，并记录相应的程序执行时间等输出。

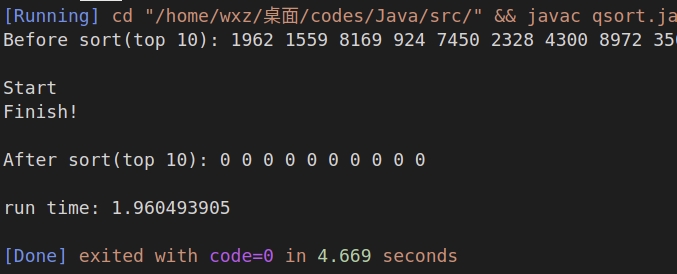
## 四、实验结果

### 1、C



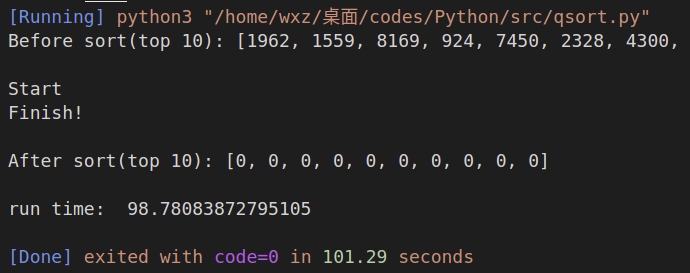
上图展示了对5000000个数字作排序的过程。代码第一行输出排序前的前10个数字，此时程序已经完成对文件的读取。Start、Finish标志快速排序函数的执行开始和结束。最后程序会输出程序的执行时间，为：3.23s

### 2、Java



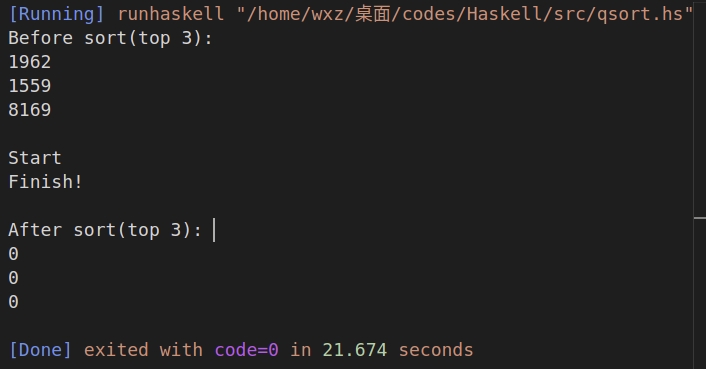
上图展示使用Java语言实现的快排对5000000个数字的排序输出结果。输出的结构与C++输出结构相同。可以看到，它的运行时间为：1.96s

### 3、Python



上图为使用Python执行排序后的结果。明显，此次排序的时间相较于前两种较长，为：98.78s。

### 4、Haskell



上图为使用Haskell执行排序后的结果。由于Haskell没有提供易用的时间模块，因此使用Visual Studio code所提供的程序运行时间作为排序的时间。这个时间是包括程序对数据文件进行读取操作的时间，因此应比实际运行时长便大。在实验中，尝试使用Haskell所提供的时间库对精确的排序程序运行时间做计算，但经实验发现此库会对程序运行效率产生极大影响，因此最终采用前述方法进行计时。由于文件操作时长较短（约为0.3s），因此使用次方法作为排序函数运行时间是可以接受的。运行时间为：21.67s。

## 五、语言易用性和程序规模对比分析

### 1、语言易用性

Python的使用最为简单，不需要考虑变量类型，使用时它会自动完成数据类型的转换和匹配。它提供的各函数及库可以简单的、忽略底层实现的完成相应的功能，因此编写程序的体验最好。对于数据读取方面，使用split方法可以方便的完成对字符串的切割。同时python的列表支持对表头、表尾的随机插入删除操作并且提供切片，因此对于列表的操作及列表输出提供了极大的便利。Python的这些特性使得它的学习较为简单，编程效率也较高。

C语言的使用相对比较复杂。它有严格的语法要求，对于变量、函数在使用前均需要进行声明，并说明其数据类型、返回值的类型。因此在使用时，需要预先对语言的语法有较为详细的了解。同时，C语言更接近机器的底层，并且提供了指针等操作。在编写程序时，需要程序设计者对及其硬件有一定的了解。同时，这也使得C语言编写程序时针对硬件进行优化的空间更大。但这也使得程序发生内存泄露的可能性增大，因此需要程序设计者有较高的设计水平。由于C语言严格的语法并且需要对硬件有一定的认识，它的学习难度较为适中。而程序的开发效率也较为适中，更多的时间可能花费在程序的调试方面

Java语言比C语言的使用稍复杂。他在对变量、函数进行使用前也需要进行声明，并说明其确定的变量类型、返回值类型。在各类库方面，这两种语言的支持也更为完善。相较下，C的类库更偏向底层，因此提供的借口更为简单和开放；而Java语言的接口使用则相对较为复杂，需要对相关的对象、操作有清晰的了解。同时在权限管理方面，由于Java是面向对象的编程语言，因此在有涉及到静态变量等共有变量时，会产生较为复杂的权限控制，需要程序编写者对Java语言的语法有清晰明确的认识。Java语言程序运行在Java虚拟机上，因此程序不直接控制计算机硬件。这使得Java程序更为安全，发生内存泄漏的概率更低，因此对程序设计者的要求也相较更低。Java语言上手较为容易，但是要对Java语言的语法特性又充分的认识需要学习者花费一定的时间进行学习。在开发效率方面，Java的类库较为丰富，因此开发效率相对较高。

Haskell语言是一种纯函数编程语言，它的程序编写较有独特的性。它的语法相比更为特殊，如没有一般性的循环，需要通过递归来实现循环功能。同时它也是强类型语言，变量在使用过程中的数据类型是明确的。在对数据与处理方面，由于Haskell的不同于一般面向过程的编程语言，因此使用起来较为困难。同时，在类库支持方面，Haskell的资料相对来说较少。而在编写快速排序程序中，Haskell语言的本身特性使得算法实现极为简洁和简单。因此Haskell语言对于科学计算、数学函数的实现较为简便。在学习方面，由于Haskell语言是函数编程语言，因此不同于其它三种程序设计语言，它的学习个人认为相对较为困难。在开发效率方面，由于自身特性，对于数学函数的实现较为容易，而对于涉及到循环及多参数函数的设计，则需要花费一定的时间。

### 2、程序规模

各个语言程序的代码行数如下：



C、Java、Python三种语言的排序实现代码部分长度基本相同，而Haskell的排序函数实现代码相较短的多。这是由于Haskell自身的语法特性，使得它对于快速排序的实现较为简单。而另外三种语言，由于通过同样的方式编写程序，因此代码长度基本相同。

对于支持部分代码，C、Java、Haskell语言的长度相似，Python语言的长度则短得多。这是由于Python语言所提供的文件操作对象简单易用，因此节约了大量的代码行数。这也使得Python的开发速度较快。

从总代码行数来看，Haskell、Python两种语言的代码长度较短，这是由他们各自的特点所造成的结果。而C、Java两种语言在代码行数上基本相同.

## 六、程序运行性能对比分析

### 1、软、硬件环境

* 操作系统：Ubuntu 18.04.4 LTS
* CPU：Intel® Core™ i5-8250U
* CPU核数：4核8线程
* CPU主频：1.6GHz
* Cache大小：三级缓存6MB
* 内存：8G

### 2、性能分析

使用5000000规模的数据作为测试数据，分别运行各语言编写的程序7次，并记录程序运行时间如下表：



从上表中可以看到，Java的执行速度最快，平均为1.94s，C语言平均速度为3.20s。而Python、Haskell的执行速度显著较慢，分别为100.82s、20.32s。

由于C、Java在执行前需要编译为可执行文件或字节码，因此可以进行充分的优化，也从而使得执行的速度更快。而Python、Haskell为解释型语言，在执行到相应的语句时才会确定接下来所要执行的内容，因此优化空间较小，执行速度较慢。同时对于Python，由于它的数组提供了更为方便用户的访问操作，因此程序访问执行时更加耗时。因此，Python、Haskell这两种解释型语言的执行速度显著慢于C、Java这两种需要编译、预编译的程序。

## 七、心得体会

本次实验中，我学习使用了4种不同的编程语言，实现快速排序这一重要算法。通过实际的程序设计语言学习、编程实践、代码运行分析，我对这四种不同的编程语言的特点有了充分的认识和了解。不同的编程语言有不同的设计思路和特点。在程序设计语言的易用性、执行速度、复杂性、安全性等方面，不同的编程语言有这不同的权衡。

以C语言来说，它更加贴近底层，可以使程序设计者有更多的空间直接控制硬件并进行充分优化，因此性能较好。但同时这也使得水平较低的程序开发者在程序设计中，可能出现内存泄漏等错误，增加了程序出现BUG的风险。而Java语言的代码需要运行在Java虚拟机上，并通过虚拟机完成内存回收等任务，有效避免了内存泄漏等错误的出现。但这样的对硬件不透明，又可能增加程序设计者对程序进一步优化的难度。Python语言使解释型语言。它相较于前两种语言，语法要求更加宽松，更加贴合人类语言的表达方式，功能使用更加方便，因此学习与编写较为简单。同时丰富的类库也使得Python可以使用较短的代码行数，完成更为复杂的任务。而这样易用性的代价，便是程序的执行效率较低。在本次实验中，Python与C语言的性能差距在30倍左右。Haskell语言的语法规则较为独特，它是一种函数式编程语言。它的语法中没有循环，需要通过递归来实现循环。因此对于一些数学公式的编程实现更为简单，更符合数学的思维方式。同时在本次编写程序过程中，Haskell语言的本身特性也使得编写的代码极为简短。但同时，它的执行效率也是最低的。

因此，对于程序语言设计者来说，需要权衡好易用性、性能、复杂性、安全性、拓展性、可移植性等多个方面的内容，这样才能设计出一款优秀的程序语言。而对于程序开发者，则需要熟悉各类程序语言的特点，根据自己的实际任务需求选择适合的程序设计语言，这样才能更加高效的完成程序设计任务。