云南大学数学与统计学院

上机实践报告

|  |  |  |
| --- | --- | --- |
| **课程名称**：近代密码学实验 | **年级**：2015级 | **上机实践成绩**： |
| **指导教师**：陆正福 | **姓名**：刘鹏 |  |
| **上机实践名称**：编程平台实验 | **学号**：20151910042 | **上机实践日期**：2018-03-07 |
| **上机实践编号**：No.01 | **组号**： | **上机实践时间**：22:39 |

# 一、实验目的

熟悉密码学编程平台和编程资源。

# 二、实验内容

1. Sage数学软件的使用，网络在线使用或下载安装使用。
2. 选做 读Java的BigInteger （java.math.BigInteger）和BigDecimal（java.math.BigDecimal）文档，分析两个类库的构成。自己构造例子熟悉BigInteger和BigDecimal中各个方法的使用
3. 在互联网查阅其它与密码学有关的编程资源，列出这些资源的网址，并予以简单介绍。

# 三、实验平台

Windows 10 1703 Enterprise（Edit Report）；

*SageMath* version 8.1, Release Date: 2017-12-07；

*Ubuntu* 17.10 x86-64（take Experiments）

*Xshell* 5 Build 1339。

# 四、实验记录与实验结果分析

1题

*SageMath*的安装与调用。

**Solution**:

SageMath是开源的软件，可以在官网上进行免费下载。由于SageMath不原生支持Windows，用虚拟机存在文件的读写问题，所以这里采用实体机安装。下载官方推荐[1]的二进制程序，免得自己编译。之后安装ssh工具，可以远程访问。具体bash命令如下

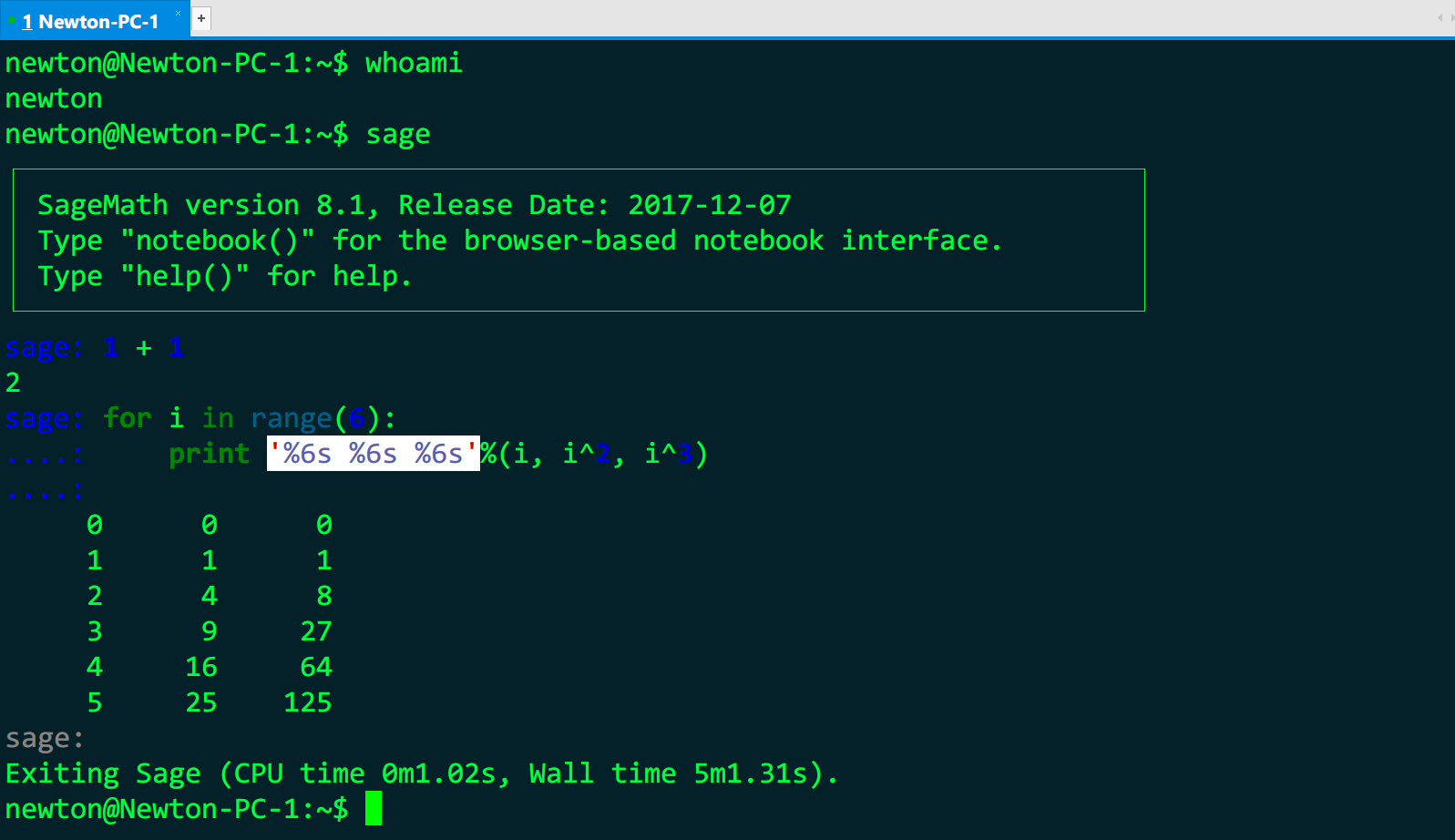
bash命令：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | newton**@**Newton-PC-1**:~/**Software**$ sudo** tar **-**zxvf sage-8.1-Ubuntu\_16.04-x86\_64.tar.bz2  newton**@**Newton-PC-1**:~/**Software**$ sudo** ln -s **~/**Software**/**SageMath**/**sage **/**usr**/**bin**/**  newton**@**Newton-PC-1**:~/**Software**$ sudo** apt-get install openssh-server  newton**@**Newton-PC-1**:~/**Software**$ sudo** **/**etc**/**init.d**/**ssh restart  newton**@**Newton-PC-1**:~/**Software**$ sudo** apt install net-tools  newton**@**Newton-PC-1**:~/**Software**$ ifconfig**  enp4s0**:** flags**=**4163**<**UP**,**BROADCAST**,**RUNNING**,**MULTICAST**>** mtu 1500  inet 192**.**168**.**1**.**78 netmask 255**.**255**.**255**.**0 broadcast 192**.**168**.**1**.**255  inet6 fe80**::**83d5**:**9345**:**5e50**:**dd58 prefixlen 64 scopeid 0x20**<**link**>**  ether 00**:**25**:**90**:**3a**:**39**:**6c txqueuelen 1000 **(**Ethernet**)**  RX packets 1067 bytes 235649 **(**235**.**6 KB**)**  RX errors 0 dropped 0 overruns 0 frame 0  TX packets 338 bytes 54874 **(**54**.**8 KB**)**  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  device interrupt 16 memory 0xfbee0000**-**fbf00000  enp5s0**:** flags**=**4099**<**UP**,**BROADCAST**,**MULTICAST**>** mtu 1500  ether 00**:**25**:**90**:**3a**:**39**:**6d txqueuelen 1000 **(**Ethernet**)**  RX packets 0 bytes 0 **(**0**.**0 B**)**  RX errors 0 dropped 0 overruns 0 frame 0  TX packets 0 bytes 0 **(**0**.**0 B**)**  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  device interrupt 17 memory 0xfbfe0000**-**fc000000  lo**:** flags**=**73**<**UP**,**LOOPBACK**,**RUNNING**>** mtu 65536  inet 127**.**0**.**0**.**1 netmask 255**.**0**.**0**.**0  inet6 **::**1 prefixlen 128 scopeid 0x10**<**host**>**  loop txqueuelen 1000 **(**Local Loopback**)**  RX packets 142 bytes 10702 **(**10**.**7 KB**)**  RX errors 0 dropped 0 overruns 0 frame 0  TX packets 142 bytes 10702 **(**10**.**7 KB**)**  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 |

程序代码 1

ssh界面

打开ssh软件，连接到Ubuntu，就可以在命令行中调用sage软件了。简单运行一下几个交互式命令。



安装过程分析：

从后期结果看，这个过程相当简单。不过在实验过程中却充满了困难。首先是平台的难度，曾经我用的是CentOS 7操作系统，因为比较稳定，而且社区比较活跃。不过CentOS 7却需要从源代码编译SageMath才可以获得可使用版本，这带来了很大的困难。先是配置编译环境很复杂，而后是编译过程很久。后来失败太多次，索性还了实验平台，Ubuntu可以很好地支持已编译的二进制文件。

2题

根据sage tutorial手册，进行重要代码的实验。

**Solution**:

Sage的基本语言是Python 2，所以之前数据结构与算法课程的很多知识可以继承过来用。Sage不像Magma，Maple，Mathematica，MATLAB等其他软件为数学重新编一种语言，而是直接可以用Python语言。Sage的后台就是一个Python解释器。虽然底层基本是Python，但是Sage的语法与Python稍有不同。Python是Sage的完整底层，而Sage在把文本交给Python之前先预处理一遍，这样就可以实现统一。这里重点关注不同之处。

（1）指数运算与异或运算

Python中的指数运算符号是\*\*，如

>>> 2\*\*9

512

但是Sage为了方便数学运算，使用了上标运算符作为幂运算符号：

sage: 2^9

512

（2）整数除法：

Python中，整数相除不会产生想要的结果，如

>>> 2/3

0

Sage在这一方面做了修正：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | sage**:** 2**/**3  2**/**3  sage**:** **(**2**/**3**).**parent**()**  Rational Field  sage**:** 2**//**3  0  sage**:** int**(**2**)** **/** int**(**3**)**  0  sage**:** |

（3）长整数

Sage使用GMP的C语言实现任意精度的整型，输出时没有L。

（4）除了少数例外，Sage使用Python语言，因此大多数关于Python的入门书籍都有助于学习Sage。

在Python中，进行不加小数点的整型运算，结果有时不满足要求，这时Sage会利用符号计算来进行求解。如果要对结果数值化，使用函数n或者方法n（两者的全名都是numerical\_approx, 并且函数N和n是一样的）。它们都有可选参数prec和digits，前者指定结果的二进制位数，即bit 数，后者指定结果的十进制位数。默认精度是53 bit。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | sage**:** exp**(**2**)**  e**^**2  sage**:** n**(**exp**(**2**))**  7**.**38905609893065  sage**:** sqrt**(**pi**).**numerical\_approx**()**  1**.**77245385090552  sage**:** sin**(**10**).**n**(**digits**=**5**)**  **-**0**.**54402  sage**:** N**(**sin**(**10**),** digits**=**10**)**  **-**0**.**5440211109  sage**:** numerical\_approx**(**pi**,** prec**=**200**)**  3**.**1415926535897932384626433832795028841971693993751058209749 |

（5）进制

Sage中，以0开头的数是表示八进制。

（6）获取帮助

Sage中，只需要输入函数或者常数的名字，再加个问号即可。

# 六、实验体会

通过这次实验，我懂得了如何配置SageMath的运行环境，同时还知道了SageMath与Python 2语言的关系。虽然这个报告尚不全面，不过在通读文档之后，可以在使用中继续学习。

# 七、参考文献

[1] 开发组 S. Sage Tutorial [M]. Release 4.3 ed., 2010.