

# 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：2018-05-27
上机实践编号：No.01	组号：	上机实践时间：08:30

### 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

Microsoft Windows 10 Pro Workstation 1803;

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

Ubuntu17.10 x86-64;

Xshell 5 Build1339。

### 四、实验记录与实验结果分析

#### 4.1 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<sup>[1]</sup>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

#### 4.1.1 bash 命令

```
1 newton@Newton-PC-1:~/Software$ sudo tar -zxvf sage-8.1-Ubuntu_16.04-x86_64.tar.bz2
2 newton@Newton-PC-1:~/Software$ sudo ln -s ~/Software/SageMath/sage /usr/bin/
3 newton@Newton-PC-1:~/Software$ sudo apt-get install openssh-server
4 newton@Newton-PC-1:~/Software$ sudo /etc/init.d/ssh restart
5 newton@Newton-PC-1:~/Software$ sudo apt install net-tools
6 newton@Newton-PC-1:~/Software$ ifconfig
7 enp4s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
```

```

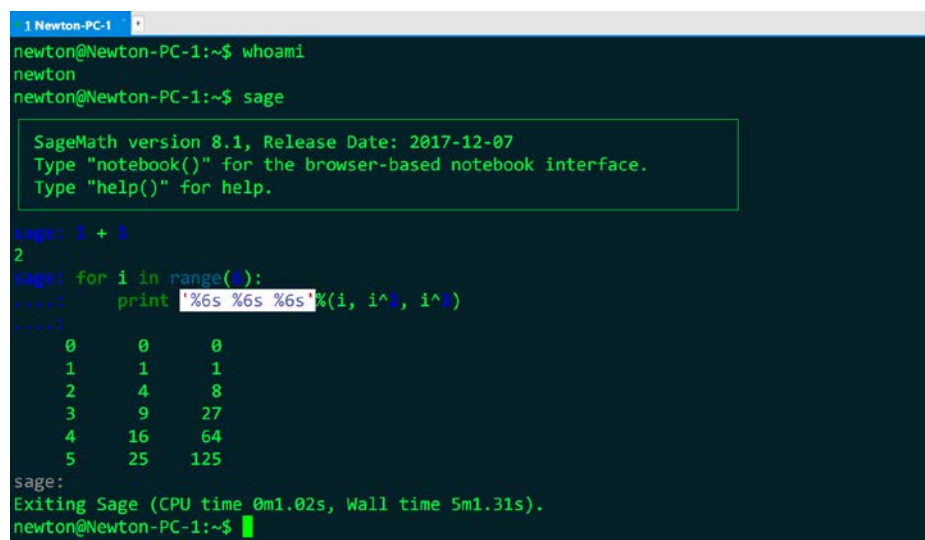
8      inet 192.168.1.78 netmask 255.255.255.0 broadcast 192.168.1.255
9      inet6 fe80::83d5:9345:5e50:dd58 prefixlen 64 scopeid 0x20<link>
10     ether 00:25:90:3a:39:6c txqueuelen 1000 (Ethernet)
11     RX packets 1067 bytes 235649 (235.6 KB)
12     RX errors 0 dropped 0 overruns 0 frame 0
13     TX packets 338 bytes 54874 (54.8 KB)
14     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
15     device interrupt 16 memory 0xfbee0000-fbf00000
16
17 enp5s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
18     ether 00:25:90:3a:39:6d txqueuelen 1000 (Ethernet)
19     RX packets 0 bytes 0 (0.0 B)
20     RX errors 0 dropped 0 overruns 0 frame 0
21     TX packets 0 bytes 0 (0.0 B)
22     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
23     device interrupt 17 memory 0xfbf00000-fc000000
24
25 lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
26     inet 127.0.0.1 netmask 255.0.0.0
27     inet6 ::1 prefixlen 128 scopeid 0x10<host>
28     loop txqueuelen 1000 (Local Loopback)
29     RX packets 142 bytes 10702 (10.7 KB)
30     RX errors 0 dropped 0 overruns 0 frame 0
31     TX packets 142 bytes 10702 (10.7 KB)
32     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

程序代码 1

#### 4.1.2 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。



```

1 Newton-PC-1
newton@Newton-PC-1:~$ whoami
newton
newton@Newton-PC-1:~$ sage

SageMath version 8.1, Release Date: 2017-12-07
Type "notebook()" for the browser-based notebook interface.
Type "help()" for help.

sage: 1 + 1
2
sage: for i in range(6):
....:     print '%6s %6s %6s'%(i, i^2, i^3)
....:
0      0      0
1      1      1
2      4      8
3      9     27
4     16     64
5     25    125
sage:
Exiting Sage (CPU time 0m1.02s, Wall time 5m1.31s).
newton@Newton-PC-1:~$

```

运行结果 1

### 4.1.3 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 CentOS7 操作系统，因为比较稳定，而且社区比较活跃。不过 CentOS7 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。

## 4.2 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 Python2，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 sage: 2/3
2 2/3
3 sage: (2/3).parent()
```

```

4 Rational Field
5 sage: 2//3
6 0
7 sage: int(2)/ int(3)
8 0
9 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 sage: exp(2)
2 e^2
3 sage: n(exp(2))
4 7.38905609893065
5 sage: sqrt(pi).numerical_approx()
6 1.77245385090552
7 sage: sin(10).n(digits=5)
8 -0.54402
9 sage: N(sin(10), digits=10)
10 -0.5440211109
11 sage: numerical_approx(pi, prec=200)
12 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。

## 五、 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 Python2 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。

另外，我发现去年十二月底，Sage 团队发布了原生 Windows 版本的软件，我的认知在完成本实验时还停留在大一阶段，所以采用了 Linux 平台。以后的实验报告均采用 Windows 原生 Sage 进行，但是这里不再更改，因为除了平台，其他方面都是相同的。

## 六、 参考文献

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