比如 Manish 上课说过的一个阶乘函数，下面这个是递归版本

```
function returnValue = fac_rec(num)
    if num == 1
        returnValue = 1;
    else
        returnValue = num * fac_rec(num - 1);
    endif
endfunction
```

下面这个是迭代版本

```
function returnValue = fac_ite(num)
    returnValue = 1;
    for factor = 1 : num
        returnValue *= factor;
    endfor
endfunction
```

一些比较小的数字，两个函数，没有多大区别。当数字增大到 100 时，递归版本已经开始出现卡顿，而迭代版本依然秒开。当增大到 300 时，运行递归版出现了报错

```
>> fac_rec(300)
error: max_recursion_depth exceeded
error: called from
    fac_rec at line 5 column 21
```

看第 2 行，这个报错不是你的程序有错，而是达到了递归的最大深度，也就是递归太多内存不够了。而迭代版依然稳如老狗。虽然会出现“`ans = Inf`”表示结果过大，程序的当前的类型已经存不下了，因此被视为了无穷大（infinite）。当然，函数还是能跑而且很快的。

抱着挑战极限的心态试了一下，大约在 `1.0e+6`（这是浮点数的科学计数法，也就是 1.0×10^6 ）时出现了明显的卡顿和风扇的抗议声（指散热风扇，不是某大佬），`1.0e+10` 时终于崩溃报错。

当然，不同机器上运行结果不一样，与处理器和内存都有关。Manish 上大课时用学校电脑上的虚拟机试时，100 左右递归版就 GG 了。

为什么递归和迭代的性能相差甚远？这个需要知道一点计算机底层的知识。当计算机运行过程中，会产生一些体积小但是需要快速访问的数据，他们会被存储在随机访问存储器（random access memory, RAM）也就是内存条中，而不是计算机的硬盘。一般来说个人计

算机的 RAM 也就 4-16 GB。而编程语言运行时，其中的一些数据会被存放到 RAM 中一个叫栈（stack）的区域。每一个函数运行时，相关的数据就会在栈中占据一定空间，行业黑话叫加了一个栈帧（stack frame）。而递归会多次调用函数，而且在被调用的函数返回值以前主调函数还不能返回值并停止，这样就导致大量的栈帧一层层的在栈中堆积，最终栈不够用了，程序就崩了。而迭代过程中，数据都只存放在几个变量中，每一次迭代这些变量值发生变化时，都会把上一次的数据覆盖而不会保留，因此迭代一般不会占用过多空间。

因此解决内存不够报错的方法，就是讲迭代改成等价的递归。这也是在实际的编程开发中迭代远比递归常用的一个原因。只熟悉递归而不熟悉迭代的同学，很抱歉，请忍痛割爱。

4.2 系统差异引发的报错

不知道你是否还记得，大明湖畔的 Coursework 2 的说明文档的 Caution 里，有这样一句话:

“*If you create functions on your own computer it is your responsibility to ensure that they perform correctly on the UNNC system. You will lose marks for functions that will NOT run properly on UNNC computers; regardless of how they will perform on your own computer.*”

我们常用的操作系统是 Windows 和 OS X，但上一届的有些学长提醒我们，学校用于验收作业的系统是 Linux，所以要警惕在自己电脑上能跑但到学校电脑不能运行的情况。此外，不少 Mac 用户也反映，Octave 在 Mac 上用户体验较差。其实 Windows 也不怎么样，因为目前 UNNC 学生的 Windows 都以 Win10 为主，然而 Octave 的安装软件和社区都有提示，实际上官方并没有正式做过 Octave 在 Win10 上的调试实验，现在的意见都是社区里的个人 Win10 用户提交的。Anyway，Octave 本身就是 Linux 世界的产物，至于 Os X 和 Windows，都是移植过去的，难免会有一些问题。所以还是建议在 Linux 的 Octave 上调试一下程序。

Linux 有非常多的发行版。目前个人用户中比较流行的是 CentOS 和 Ubuntu。开始我比较推荐 CentOS，因为以后如果有同学需要学习搭建 Linux 服务器的话 CentOS 比 Ubuntu 更稳定专业、更适合做服务器，而且和 Manish 是同款系统 [滑稽]。然而最近发现，CentOS 官网上几个镜像下载下来的 CentOS 7 都是不自带图形化界面（GUI）的，对于新手来说自

己装图形化界面有点麻烦。而且网上查到的个在 CentOS 上安装 Octave 方法，下载下来的 Octave 都是不带图型化界面的。所以，如果你和我一样是 Linux 小白，而且目前就是急着调试 Octave 的话，我推荐 Ubuntu。而且刚才说 Ubuntu 不太专业也是和有一些 Linux 系统相比，和 Win 以及 OS X 相比当然还是更适合编程开发。如果有大佬知道怎么在 CentOS 下载有 GUI 的 Octave 的话麻烦分享一下。

至于怎么下载虚拟机和安装 Linux，这里有一篇比较好的文章：

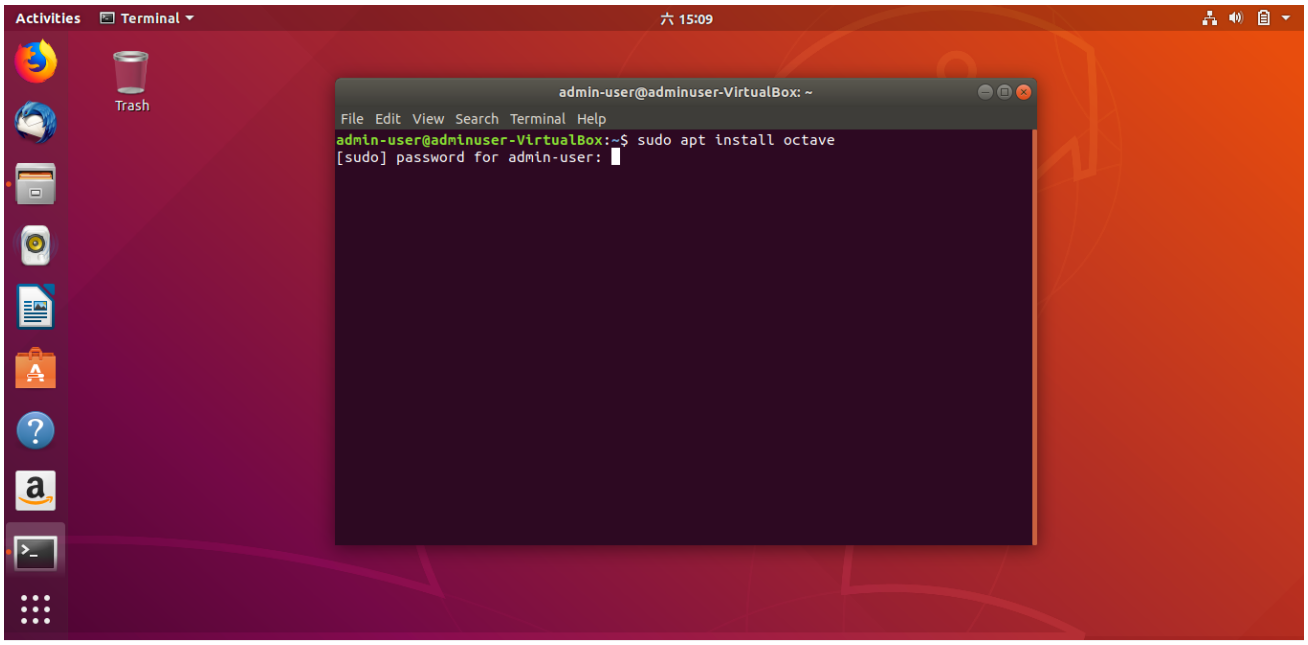
`https://blog.csdn.net/haoxiangtianxia/article/details/19993977`

在此不再赘述了。但是这个教程中有一点我不太赞同，就是给 Ubuntu 分配 1G 的内存，亲测只有 1G 内存 Ubuntu 很容易卡死，建议至少 1.5G，但也不要超过实际计算机内存的一半。还有，如果你是 Windows 用户，虚拟机默认会把虚拟盘放到 C 盘，那个虚拟盘能最大到 10 多个 G，如果你的 C 盘不大，建议自己选择一个合适的路径放虚拟盘。另外，在去官网下载 Ubuntu 的光盘映像时，注意看一下你的机器是 32 位（x86）还是 64 位（x64），如果是 32 位那么你就不能装 64 位的系统。

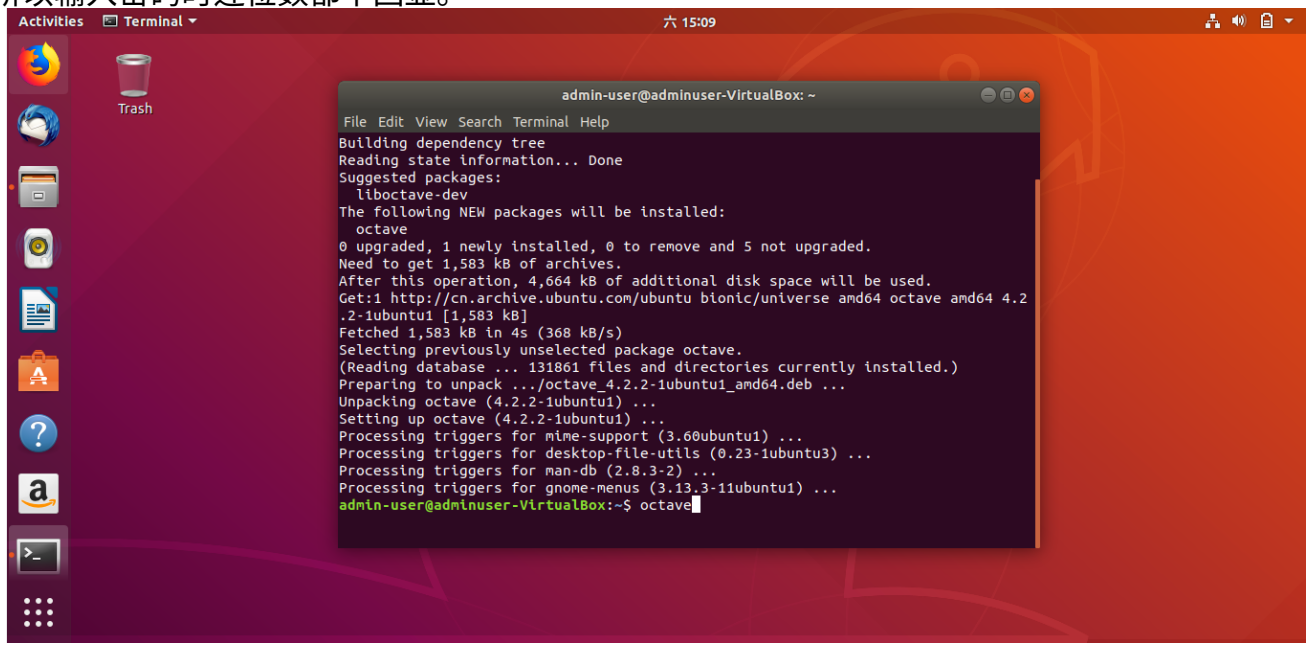
安装好 Ubuntu 后，点击屏幕左下角，有点相当于安卓的“所有应用”。然后在里面找到终端（Teminal），点开。

在里面输入

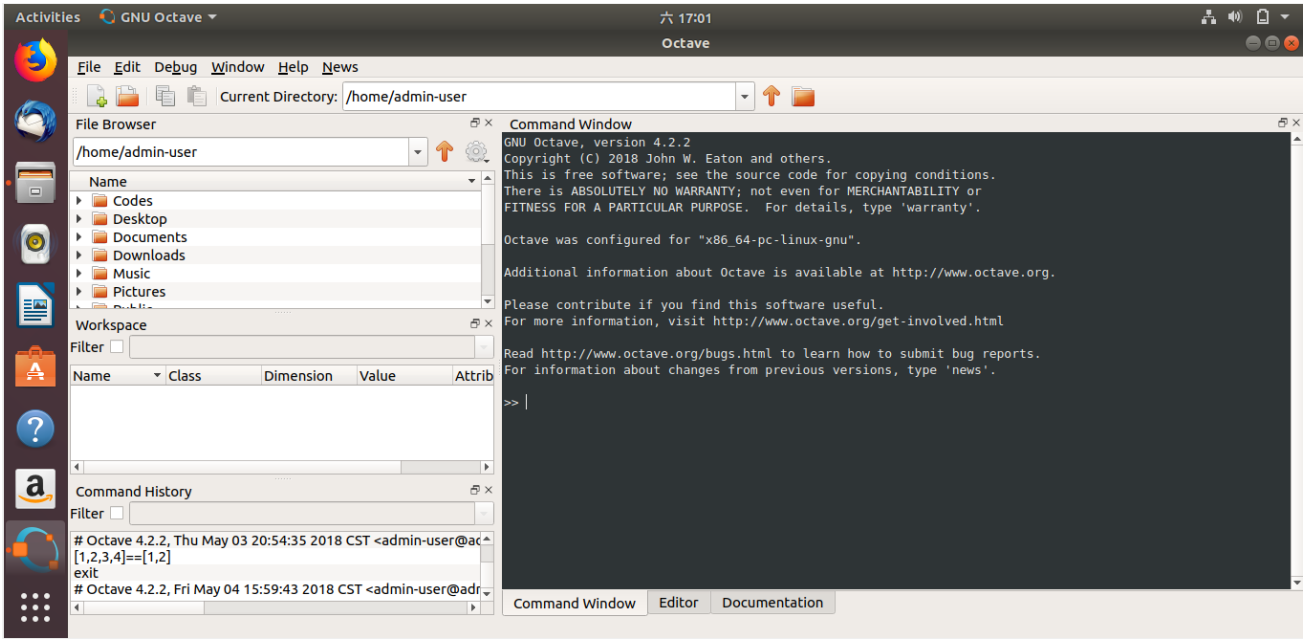
```
1 sudo apt install octave
```



敲回车，然后让你输入刚才装系统时你给自己账户设置的密码（Linux 中密码很重要，很多事情都要输入密码，瞎设密码导致忘了的话就凉了）。然后你会发现好像程序和卡死了一样，输入密码也不出现“***”什么的。这其实是 Linux 保护用户密码的机制，为了防止密码被偷窥所以输入密码时连位数都不回显。



然后屏幕上就会打印出各种安装状态，也可能会向你确认是否安装。然后一切都安装好后，再在里面输入 `octave`，或者再去那个“所有应用程序”的界面找 Octave 的图标。



然后就可以和 Windows 和 Mac 上一样愉快的玩耍了。

[本期完]