一．

1. 简单的说是计算机内部用来存储浮点数的字节数有限，位数多了就存不下了，比如111111111110.1111111111111111111111。也不一定是位数多(有多于 99.999...% 的数在计算机中是 **不能表示** 的。 真的是太令人吃惊，也太令人遗憾了。 真相总是很残忍) 比如0.1到0.9的9个十进制小数中，只有0.5可以用二进制精确表示，想知道为什么自己转换算算吧哈哈。

详细的浮点数存储知识请看IEEE 754.

2.**递归算法：**

优点：代码简洁、清晰，并且容易验证正确性。

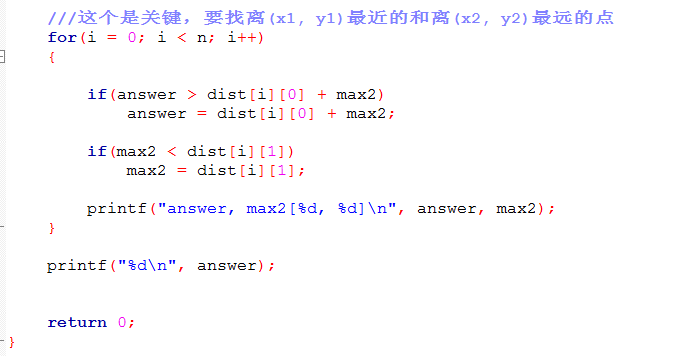
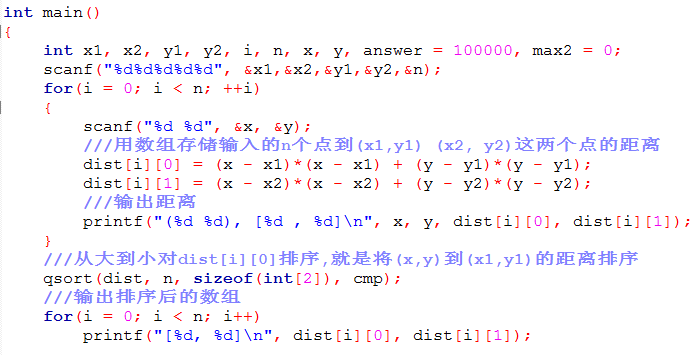
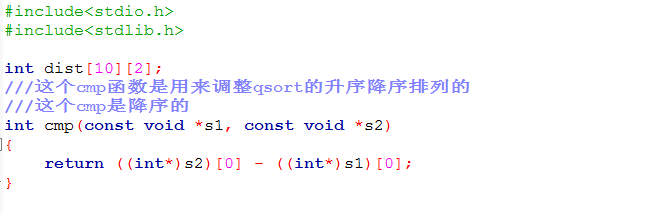
缺点：它的运行需要较多次数的函数调用，如果调用层数比较深，需要增加额外的堆栈处理，比如参数传递需要压栈等操作，会对执行效率有一定影响

**循环：**

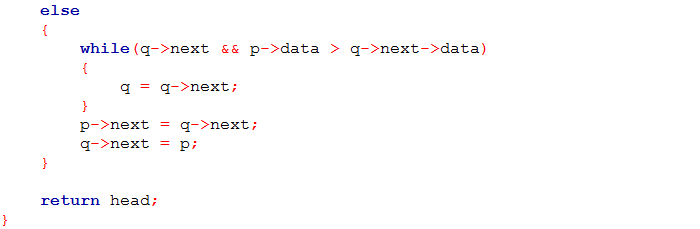
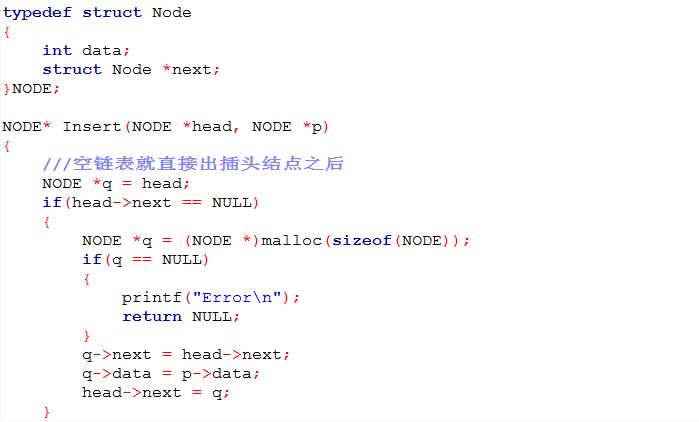
优点：速度快，结构简单。

缺点：并不能解决所有的问题。有的问题适合使用递归而不是循环。如果使用循环并不困难的话，最好使用循环。

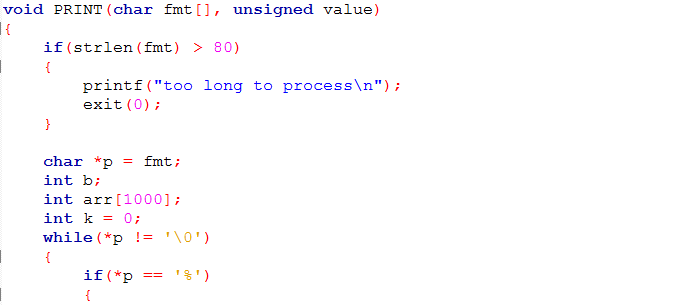
二．每年读程序题都有点绕，且看程序



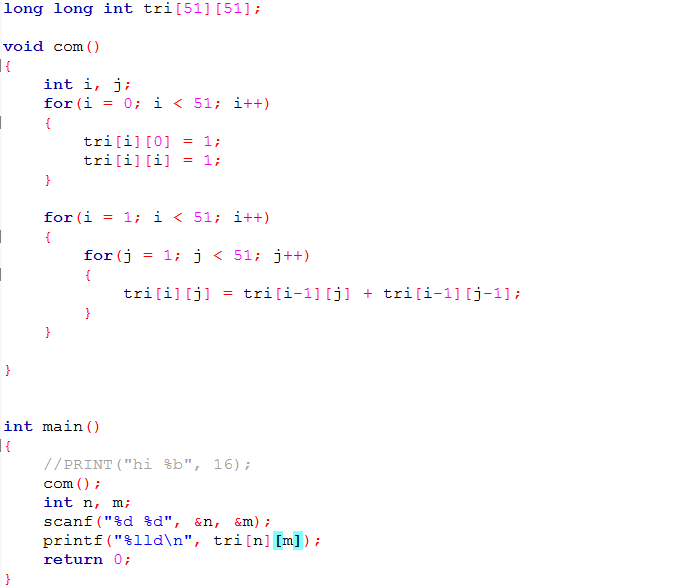
三、（1）链表插入



（2）格式化输出

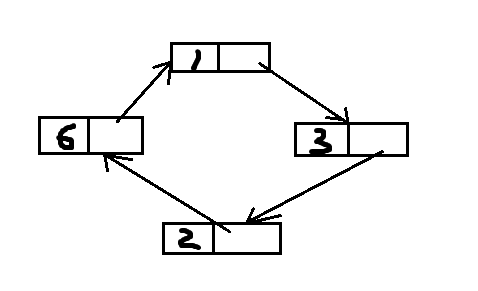


1. 组合数，用二维数组存储就行了

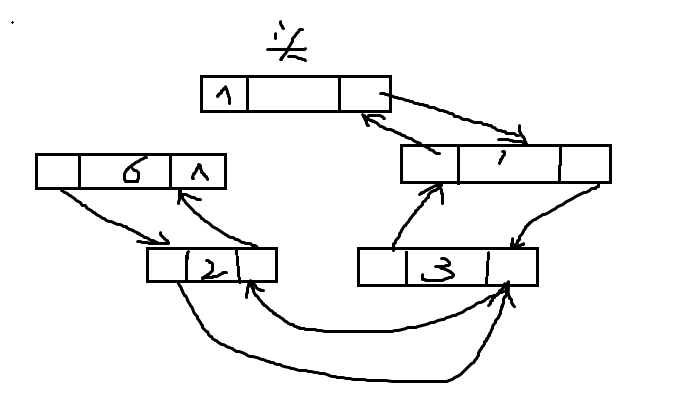


数据结构部分

1.（1）



（2）



2 广义表的画法有两种（参照严书，电脑画图丑不画了）

3. 5 8 2 9 6

选择排序 快排

1） 2 8 5 9 6 1）2 5 8 9 6

2） 2 5 8 9 6 2）2 5 6 8 9

3） 2 5 6 9 8

4） 2 5 6 8 9

4.

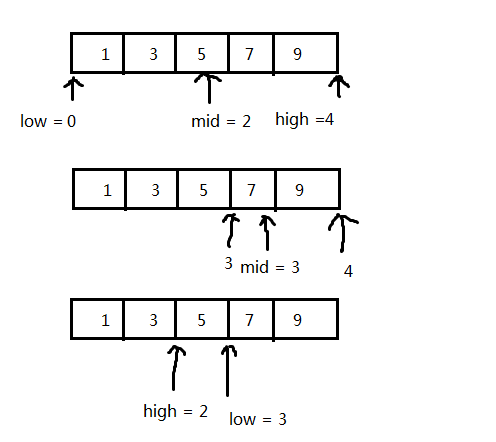
1) 行序 t[i][j] = t[0][0] + (i\*7 + j)\*s

列序 t[i][j] = t[0][0] + (j\*2 + i)\*s

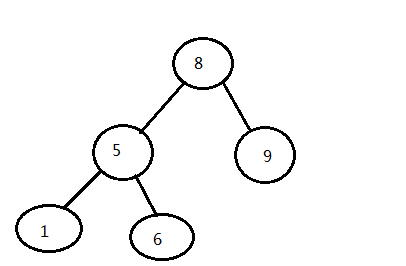
2) 十字链表的也是看书就知道了

5

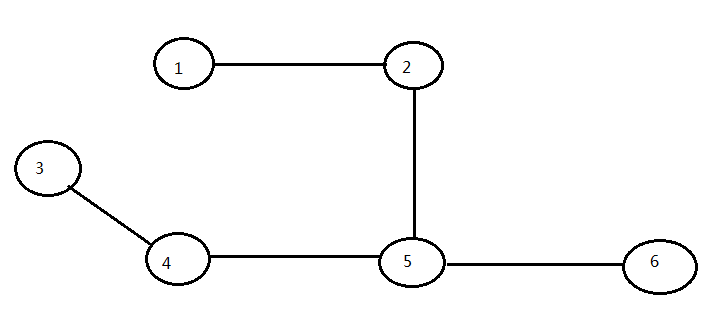
1）二分查找 6

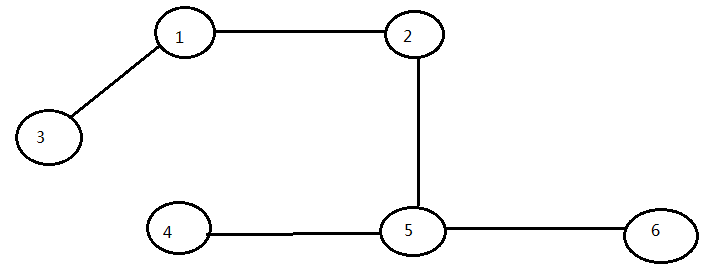


1. 平衡查找树



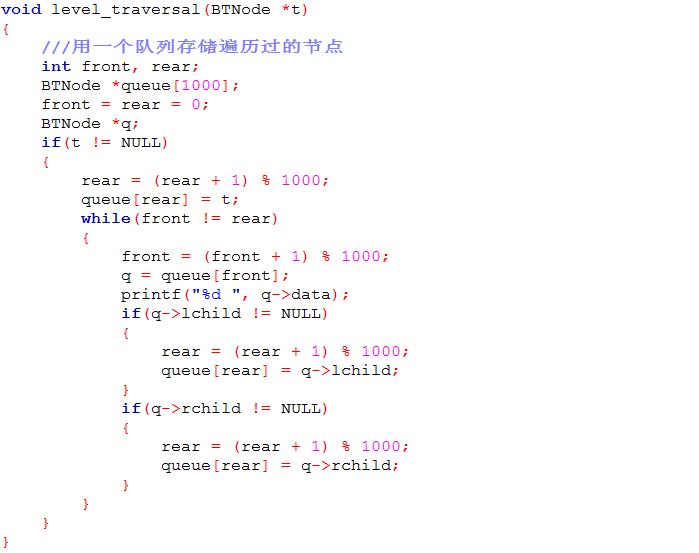
6. 1） 1 2 5 6 4 3



2

7

1. 层次遍历



1. 判断无向图连通性

在对无向图进行遍历时，对于连通图，仅需从图中任一顶点出发，进行深度优先搜索或广度优先搜索，便可访问到图中所有顶点。

把图遍历一次，就知道是不是连通的了。代码太长不放这了我传github上面去