06.27 红黑树，B树和B+树.

红黑树是二叉树，大规模数据存储时红黑树很深，所需IO很多，效率低下，对存储设备寿命的影响也很大。因此，我们需要使用多叉的搜索树，来降低树的高度，提高效率。

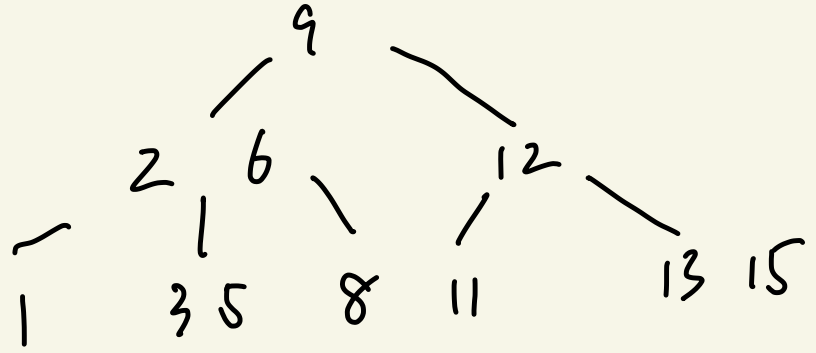
B树：m叉的搜索树，m称为B树的阶，m取决于磁盘页的大小。$m>2$.

~~根结点的孩子树：$[2,m]$~~

~~非根非叶结点的孩子数：$[m/2,m]$~~

非叶结点的关键字个数(k-v对)：孩子数-1.

k个关键字个数把节点拆成k+1段，指向k+1个孩子，



所有叶子节点位于同一层。

与B+树不同的点：关键字分布在整棵树

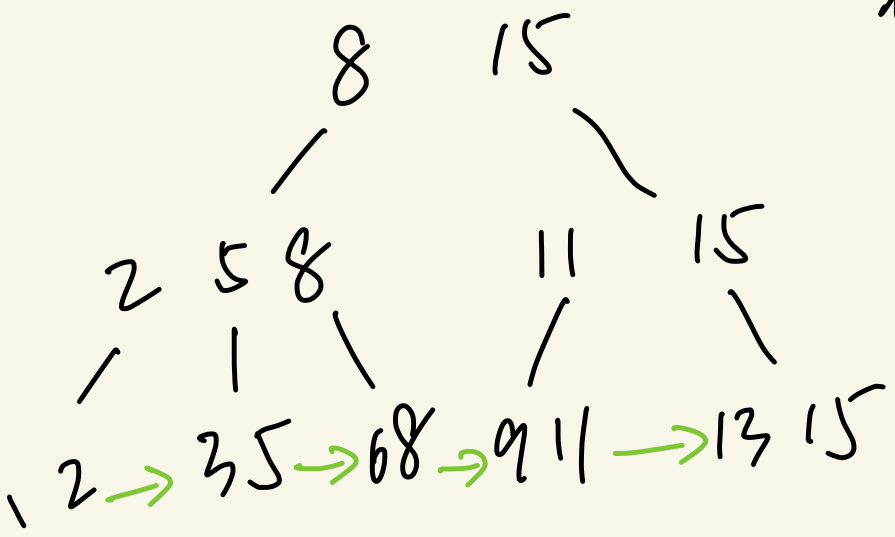
任何关键字只出现一次

搜索可能停止在非叶节点

B+树：一个节点K个关键字，拆成K段，指向K个孩子

（B树会差1）。 根结点至少

关键字是B树的最大元素，一些关键字是出现多次的。



非叶节点不存储V，只作索引

叶节点才存K-V

所有叶节点用链表相连，range-query方便

子树个数，关键字个数：

B：每个节点最多有m个子树；根节点最少2个子树，其他非叶节点最少ceil(m/2)个子树。非叶节点：关键字个数=子树个数-1.

B+：子树个数的情况和B树一样，关键字数=子树个数

B+树优点：非叶节点不存储V的信息，进一步压缩空间，m可以进一步变大。range-query方便（B树找数比中序遍历）

B树：可能可以不到叶节点就找到，更快。（B+树：只能到叶节点找到V，但是IO性能更稳定）

06.26 Python内存管理特点

1. 引用计数

Python内部记录3对象有多少个引用。对象创建时会创建一个引用计数，当引用计数为0，会被GC.

① 分配新名称/放入容器，引用计数增加

② del/引用超出作用域/重新赋值 引用计数减少

③ sys.getrefcount() 获取引用计数

2. GC：检查引用计数为0的对象，清除内存空间

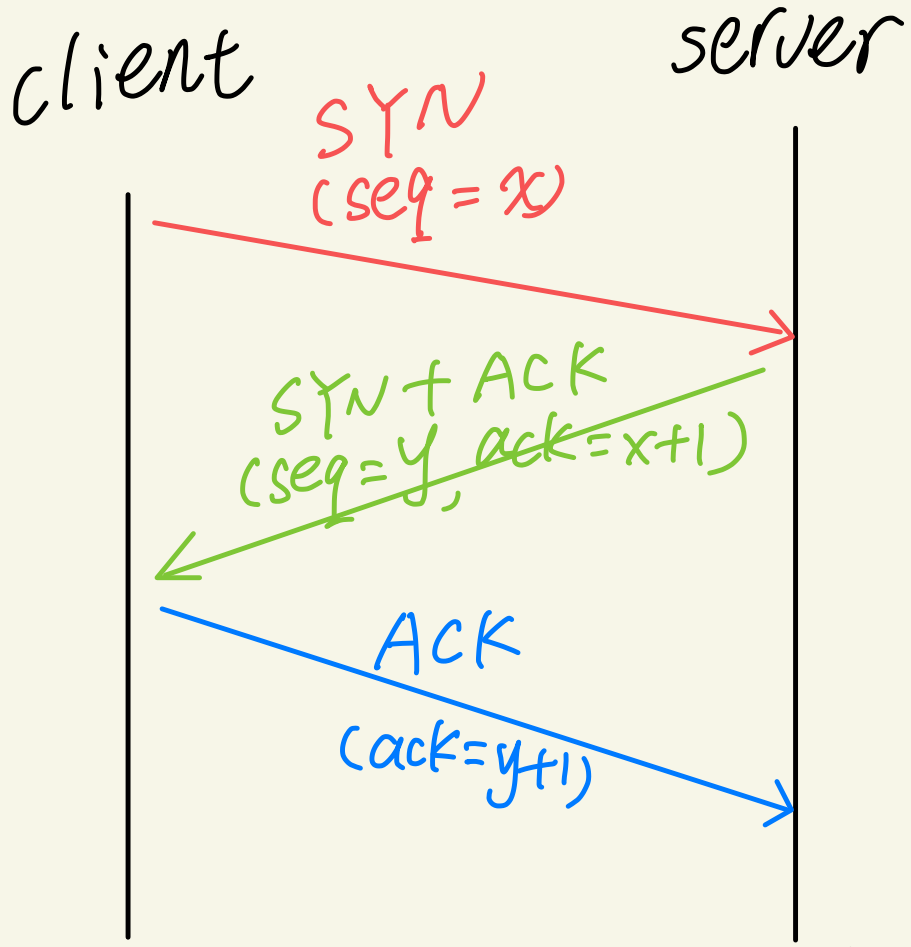
3. 内存池：很多小内存申请释放频繁，会影响效率。引入内存池机制，用于管理小块内存。能够减少内存碎片。

① GC将内存返还给内存池而非操作系统

② 小于256B的对象使用Pymalloc分配器，大的使用系统malloc

③ Int, Float, List等之间不共享内存池

06.23 TCP 3-way handshake process



TCP usea the 3-way handshake process to establish a secure and reliable communication between the client and server.

① client sends the SYN to the server (synchronization) with SYN Flag as 1, means client wants to establish a secure connection with the server. seq = x, which is a random number, assigned to the first bit of the data.

② Server receives the SYN from client, sends SYN + Ack to the client, with SYN and Ack Flag as 1. ack number is set as x+1, tell client that the server has successfully receives the previous SYN from the client. Anothe seq is set as y, which is a random number, assigned to the first bit of the data.

③ Client receives the SYN+ACK from the server, send an ACK to the server, with ack number set as y+1.

After the 3-way handshake, the TCP connection is established.

06.22

oop features,

06.21 Describe what happens when you click on a URL in a browser

① You type the URL, for example, www.leetcode.com into the browser.

② The browser will contact the DNS (domain name system), to convert the human-readable URL into the numerical IP address.

③ With the IP address, the browser can send an HTTP request to the server.

④ If the server receives the HTTP request from the browser, it will prase

the request, and send an HTTP response to the browser.

⑤ After receiving the response, the browser will render the HTML encoded in the response. If there are additional rescourses embedden in the HTML, steps 3~5 will be repeated.

4 layer cache:

browser, os, router, ISP

recursively request the IP address

Root DNS

Top-level .com

Second-level microsoft.com

Third-level download.microsoft.com

TCP connection, 3-way handshake