## 2020/08/03==================================

1. 自我介绍
2. 哈希表怎么实现，冲突怎么解决
   1. 哈希表的底层数据结构是数组，很多地方也叫Bucket。首先通过将key的值传给hash函数，求出对应的索引，找到相应的下标进行存储，时间复杂度是O（1）。
   2. 解决方法：
      1. 开放定址法
      2. 再hash法
      3. 链地址法（HashMap）
3. 维护一个堆（logn）
4. Nlogn
5. B树和B+树
   1. B树是多路平衡搜索树，它类似于普通平衡二叉树，区别是允许每个节点有多个子节点。B树为外部存储器（读写磁盘）设计，用于读写大块数据。
   2. 空间局部性原理：存储器的某个位置被访问，它附近的位置也被访问。
   3. B+树
      1. 叶子节点存储数据，非叶子节点并不存储数据
      2. 叶子节点增加了链指针
      3. 因为不存data，所以B+树叶子节点大小不同，B树基本相同。
   4. 区别
      1. B树的非叶子节点保存key和value，而B+树的非叶子节点只保存key的副本，叶子节点保存value（data值）。B+树查询时间复杂的logn，B树则与位置有关。
      2. B+树叶子节点数据是用链表连起来的，可以做到区间访问性，访问磁盘某个位置，附件位置也被访问。
      3. B+树适合外部存储，key小，磁盘单次IO信息量大，IO次数少。
   5. Mysql的数据结构是B+树