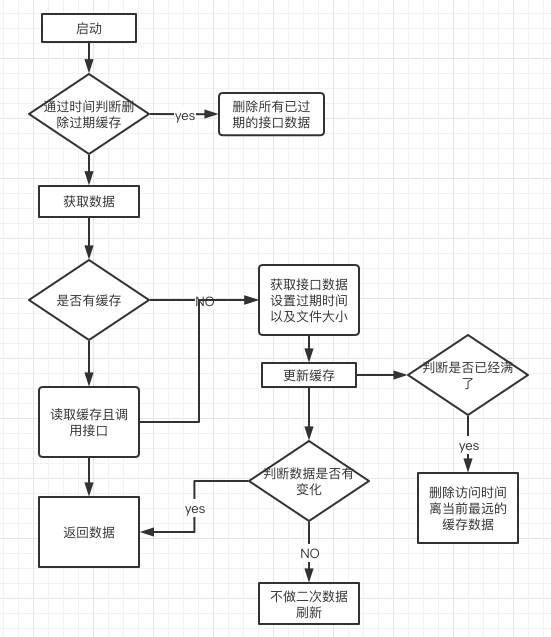
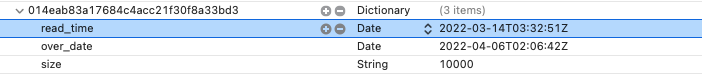
关于接口缓存的LRU实现

LRU的理论：

1. 利用一个链表来实现，每次新插入数据的时候将新数据插到链表的头部；每次缓存命中（即数据被访问），则将数据移到链表头部；那么当链表满的时候，就将链表尾部的数据丢弃。
2. 用一个数组来存储数据，给每一个数据项标记一个访问时间戳，每次插入新数据项的时候，先把数组中存在的数据项的时间戳自增，并将新数据项的时间戳置为0并插入到数组中。每次访问数组中的数据项的时候，将被访问的数据项的时间戳置为0。当数组空间已满时，将时间戳最大的数据项淘汰。
3. 利用链表和hashmap。当需要插入新的数据项的时候，如果新数据项在链表中存在（一般称为命中），则把该节点移到链表头部，如果不存在，则新建一个节点，放到链表头部，若缓存满了，则把链表最后一个节点删除即可。在访问数据的时候，如果数据项在链表中存在，则把该节点移到链表头部，否则返回-1。这样一来在链表尾部的节点就是最近最久未访问的数据项。

自己缓存的实现：  
1、流程图：

  
2、数据保存格式  
  
3、主要实现过程。Key为通过url去除请求头和部分不必要数据，然后md5加密。数据已文件方式保存。将response 直接转换为data 然后以key值文文件名进行保存。同时保存缓存的plist，也是以数据格式如上图。而且为了加快读写速度在每次读取到缓存时直接用字典保存缓存数据下次读取时直接冲内存中读取，不需要再从文件读取。

iOS 数据存储方式  
