# 跳表

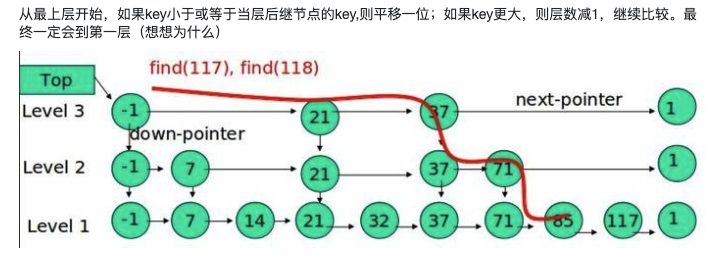
skiplist和各种平衡树（如AVL、红黑树等）的元素是有序排列的，而哈希表不是有序的。因此，在哈希表上只能做单个key的查找，不适宜做范围查找简单。所谓范围查找，指的是查找那些大小在指定的两个值之间的所有节点。

平衡树的插入和删除操作可能引发子树的调整，逻辑复杂，而skiplist的插入和删除只需要修改相邻节点的指针，操作简单又快速。

从内存占用上来说，skiplist比平衡树更灵活一些。一般来说，平衡树每个节点包含2个指针（分别指向左右子树），而skiplist每个节点包含的指针数目平均为1/(1-p)，具体取决于参数p的大小。如果像Redis里的实现一样，取p=1/4，那么平均每个节点包含1.33个指针，比平衡树更有优势。

红黑树,每个节点都固定有两个子节点,而跳表是可以自己指定的.

redis也是为了内存小才用的跳表.



可以看到就是先从上面来,发现已经大于我了,那么我去下一层找.

上面的几层可以看作是索引.

你看啊基本就是和红黑树差不多啊,上面小,下面大.