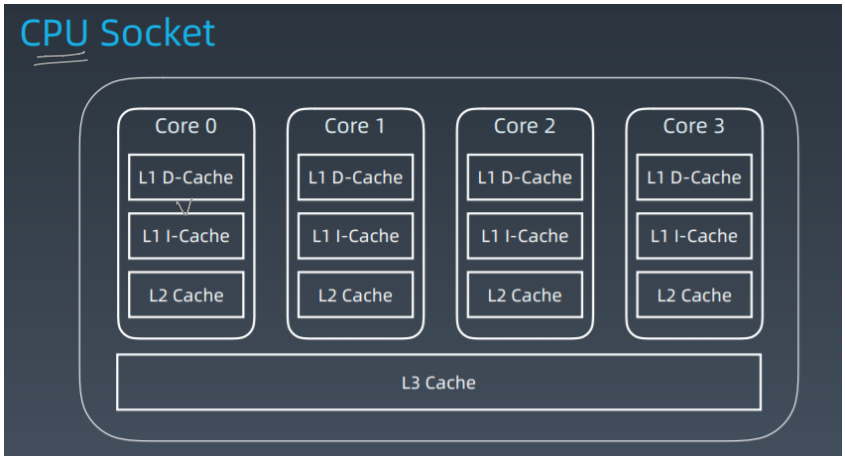
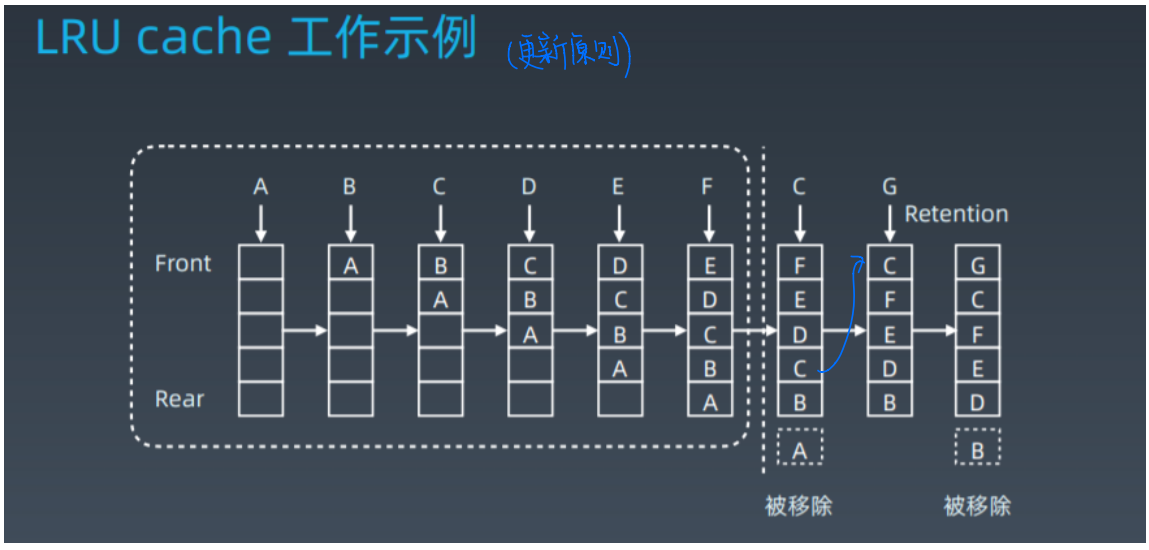
Cache 缓存

举例 1)记忆 2)钱包 - 储物柜 3)代码模块



L1 最快 容量有限

缓存非常大，类似于 CPU的缓存抵一个内存



替换策略 LRU：Least Recent Used 最近最少使用的

LFU：Least Frequently Used

(还要统计每个元素被使用到的频次最少的)

频次最少的，放在最下面，最先被淘汰掉)

...

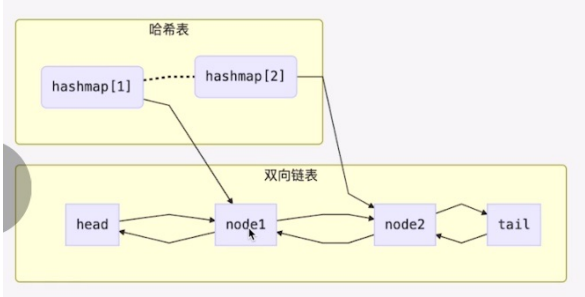
面试

LRU 是最近最少使用的算法的意思

(可以调用 API)

哈希表 + 双向链表 ☆

Java 也可以用 LinkedHashMap



→ 用哈希表来存相应一个元素，

它所指的链表所在的位置

置好了之后 node1 假设要删除，

找到它的前驱 & 找到它的后继，

后继的前驱放到前面来

插入 → 从 head 这里插进去即可

LRU Cache — Java

```java
public class LRUCache {

    private Map<Integer, Integer> map;

    public LRUCache(int capacity) {
        map = new LinkedCappedHashMap<>(capacity);
    }

    public int get(int key) {
        if(!map.containsKey(key)) { return -1; }
        return map.get(key);
    }

    public void put(int key, int value) {
        map.put(key,value);
    }

    private static class LinkedCappedHashMap<K,V> extends LinkedHashMap<K,V> {
        int maximumCapacity;

        LinkedCappedHashMap(int maximumCapacity) {
            super(16, 0.75f, true);
            this.maximumCapacity = maximumCapacity;
        }

        protected boolean removeEldestEntry(Map.Entry eldest) {
            return size() > maximumCapacity;
        }
    }
}
```

→ 看 Leet Code 官方题解