

词典

排解冲突：懒惰删除

11 - C3

邓俊辉

deng@tsinghua.edu.cn

江山故宅空文藻，云雨荒台岂梦思
最是楚宫俱泯灭，舟人指点到今疑

故居 ~ 空宅

❖ Lazy Removal :

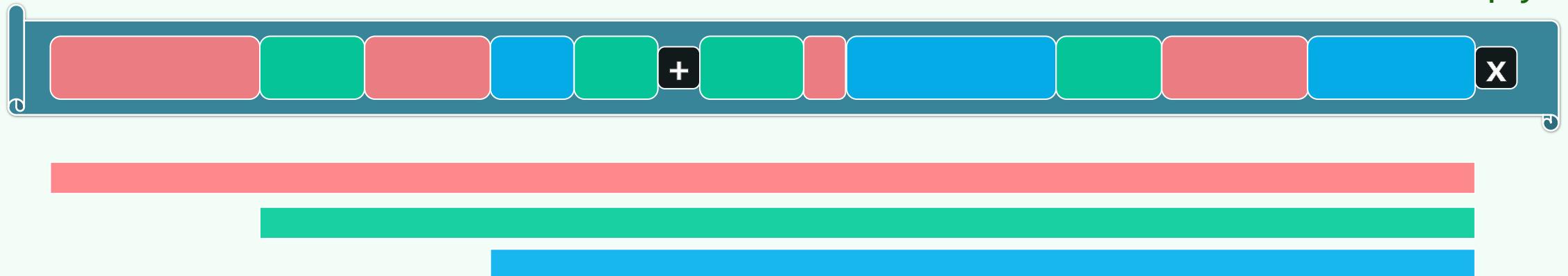
仅做标记，不对查找链做更多调整

故居

空宅

LM

Empty



❖ 此后，带有删除标记的桶所扮演的角色，因具体的操作而异

- 查找词条时，被视作“**必不匹配的非空桶**”，查找链在此得以延续
- 插入词条时，被视作“**必然匹配的空闲桶**”，可以用来**存放新词条**

算法

```
❖ template <typename K, typename V> int Hashtable<K, V>::probe4Hit(const K& k) {  
    int r = hashCode(k) % M; //从首个桶起沿查找链，跳过所有冲突的和被懒惰删除的桶  
    while ( ( ht[r] && (k != ht[r]->key) ) || ( !ht[r] && lazilyRemoved(r) ) )  
        r = ( r + 1 ) % M; //线性试探（注意并列判断的次序，命中可能性更大者前置）  
    return r; //调用者根据ht[r]是否为空，即可判断查找是否成功  
}  
  
❖ template <typename K, typename V> int Hashtable<K, V>::probe4Free(const K& k) {  
    int r = hashCode(k) % M; //从首个桶起  
    while ( ht[r] ) r = (r + 1) % M; //沿查找链找到第一个空桶（无论是否懒惰删除）  
    return r; //调用者根据ht[r]是否为空，即可判断查找是否成功  
}
```