

词典

排解冲突：封闭散列

旅客要在每个生人门口敲叩 才能敲到自己的家门
人要在外面到处漂流 最后才能走到最深的内殿

在我们出生之前，一切都在没有我们的宇宙里开着
在我们活着的时候，一切都在我们身体里闭着
当我们死去，一切重又打开
打开、关闭、打开，我们就是这样

一俊遮百丑

邓俊辉

deng@tsinghua.edu.cn

公共溢出区 / Overflow Area

❖ 单独开辟一块连续空间

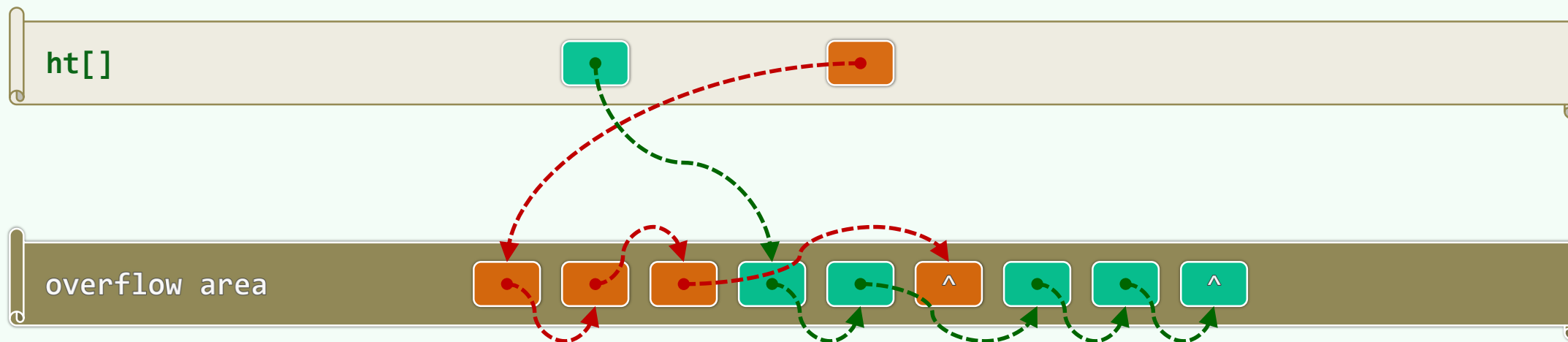
发生冲突的词条，顺序存入此区域

❖ 结构简单，算法易于实现

❖ 但是，不冲突则已，一旦发生冲突

最坏情况下，处理冲突词条所需的时间将

正比于溢出区的规模

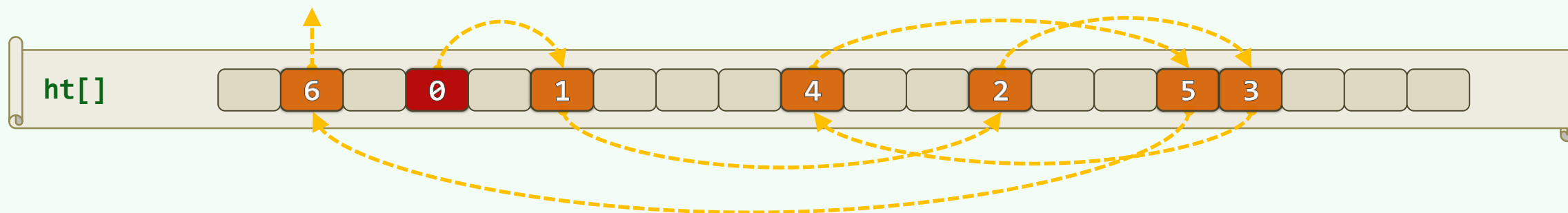


封闭散列 x 开放定址 x 试探链

❖ Closed Hashing: 散列表的物理空间相对**固定**；所有冲突都在**内部**寻求解决

Open Addressing: 只要有**必要**，任何散列桶**都可以**接纳任何词条

❖ 与开放散列恰好**相反**，整体结构更加整饬、简明，cache机制可大显身手



❖ Probe Sequence/Chain: 为每一组同义词，都**事先约定**了若干**备用桶**，依次串接成序列/链

❖ 查找算法: 沿**试探链**逐个检视**下一桶单元**，直到命中**成功**，或者抵达一个**空桶而失败**

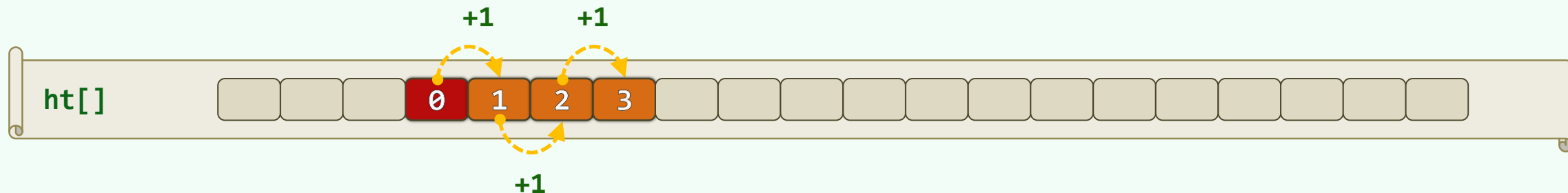
❖ 具体地，试探链又应如何**约定**？

线性试探

❖ Linear Probing: $r_i(key) = (\text{hash}(key) + i) \bmod \mathcal{M}, \quad i = 0, 1, 2, 3, \dots$

❖ 当然，只要还有**空桶**，试探链就不致**循环**，查找必能**终止**

然而，试探链有相互**重叠**，**非同义词**之间也会彼此**堆积** (clustering)



❖ 但只要**控制好**装填因子，冲突与堆积都**不致**太严重: $\mathbb{E}(\text{probes}) = 1/(1 - \lambda)$

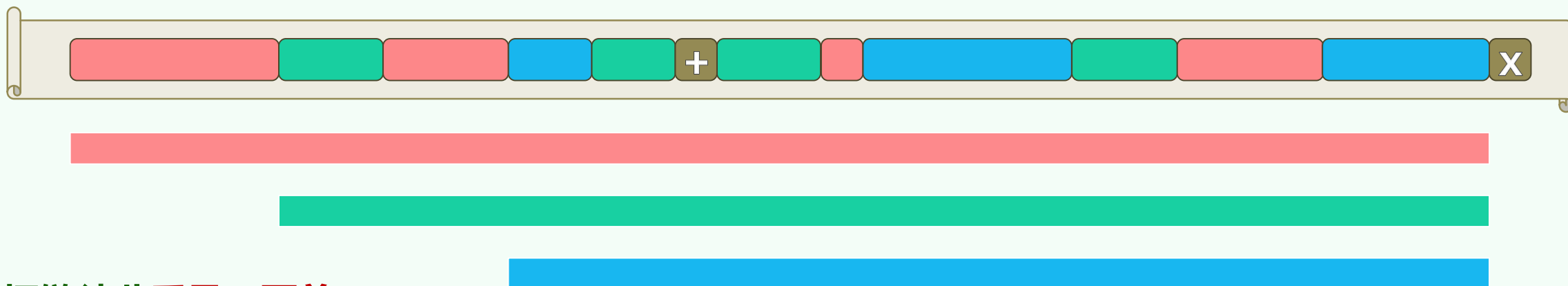
❖ 更重要地，**一俊遮百丑**...这是典型的Stride-1 Reference Pattern

数据局部性**极佳**，系统**缓存**的作用可发挥到**极致**

插入 + 删除

❖ 插入：新词条若尚不存在，便可存入试探**终止**处的空桶

❖ 删除：简单地将命中的桶**清空**？不可！



❖ 试探链彼此**重叠、覆盖**！

一个桶的清空，将令**多条**试探链在此**断裂**，导致后续的词条**丢失**——明明存在，却访问不到

❖ 那么，如何才能**简明、高效而安全**地完成删除呢？