

映射

$0-10^9 \rightarrow 0-10^5$

$h(n)$

哈希函数

① 哈希函数如何写? $A \bmod 10^5$

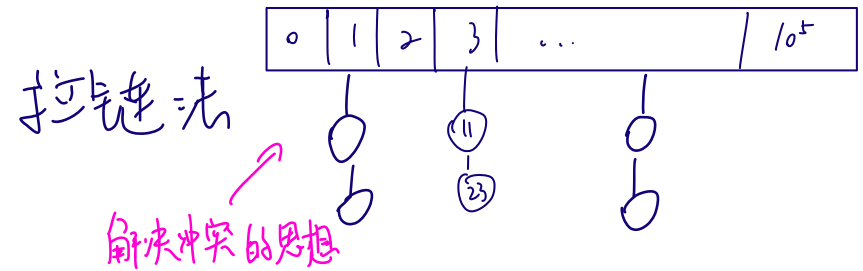
② 解决冲突

取值数

若干不同的数 映射 相同的数

(离散化 \rightarrow 特殊的哈希)

\downarrow 要保序



1. 添加 2. 查找 3. 打标(删除)

存一串 head

开放寻址法



只开一个数组: 2~3倍长度

\rightarrow 找坑位

1. 添加 2. 查找 3. 删除(打标)

find { 存在: 位
不存在: 应存位

字符串前缀哈希法

str = "ABC ABCDEF ..."

h[0] = "0" 的哈希

h[1] = "A" 的哈希

h[2] = "AB" 的哈希

"ABCD"

⇒ A B C D

(1 2 3 4) p

进制

$$= (1 \times p^3 + 2 \times p^2 + 3 \times p + 4 \times p^0) / 10$$

(比较大)



mod Q

① 不能映射为0

A → 0

AA → 0

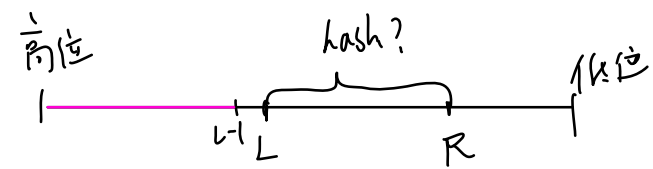
... 冲突

② 假设RP足够好, 不存在冲突

$$\begin{cases} p=131 \text{ 或 } 13331 \\ Q=2^{64} \end{cases}$$

77.77% 不冲突

相当于补位: 再相减, 获得 L-R 段哈希



用处: 前缀哈希 → 子串哈希

已知: h[R]
h[L-1]

$$h[R] - h[L-1] \times p^{R-L+1} = \text{hash?}$$

Trick: Q - mod 2⁶⁴

用 unsigned long long 存储 → 溢出不用取模

作用: 询问两段字符串
(是否相同)

STL

- vector
- string
- queue, priority-queue
- stack
- deque
- set, map, multiset, multimap
- unordered-set, unordered-map
- bitset

1. Vector: 变长数组 (倍增)
2. String: 字符串, substr(), c-str()
3. queue: 队列, push(), front(), pop()
4. priority-queue: 优先队列: push(), top(), pop()
5. deque 双端队列
6. set, map, multiset, multimap 基于平衡二叉树, 动态维护有序序列
7. unordered-set, unordered-map 哈希表
8. bitset 压位