10 检索

定义

- 平均检索长度 (ASL)

$$ASL = \sum_{i=1}^{n} P_i C_i$$

> 注意概率分布的隐含条件 $\sum p = 1$

方法

线性表检索

"监视哨"顺序检索算法

- 即为在待检索的数组末尾加上 $a[n] = key$, 以节省遍历时的数组越界检查, 复杂度还是 $O(n)$
- 性能分析: (假设检索成功率是 p)

$$\frac{n+1}{2} < ASL = \frac{n+1}{2}p + (n+1)(1-p) < n+1$$

- 优点 : 插入元素可以直接加在末尾
- 缺点 : 检索时间太长 $\Theta(n)$

二分检索法

- 就是二分法, 需要先对数据排序
- 成功的平均检索长度

$$ASL_{succ} = \frac{1}{n}\sum_{i=1}^{j} i \cdot 2^{i-1} = \frac{n+1}{n}\log_2(n+1) - 1 \approx \log_2(n+1) - 1$$

- 最大检索长度 $\lceil \log_1(n+1) \rceil$, 失败检索长度 $\lceil \log_2(n+1) \rceil$ 或 $\lfloor \log_2(n+1) \rfloor$
- 优点: 平均与最大检索长度相近, 检索速度快
- 缺点: 要排序, 顺序存储, 不易更新(插/删)

分块检索

- 按块有序, 块内无序, 是顺序与二分法的折衷
- n 个元素, b 个块 , 每个块至多 s 个元素

$$
\begin{aligned}
ASL &= ASL_b + ASL_w \\
&\approx \log_2(b+1) - 1 + \frac{s+1}{2}
\end{aligned}
$$

- 当大量插入/删除时, 或结点分布不均匀时, 速度下降

集合检索

- 对于密集型集合, 用位向量表示, $\{0,1\}$ 表示是否有这个元素
  - 用 ulong 存, 每个存32个位
  - 位运算模拟集合交并

散列表检索

- **负载因子** $\alpha =$ 填入表中的结点 $n/$ 散列表空间大小 m
- **冲突** : 不同关键码计算出了相同的散列地址
  - 现实中不冲突的散列函数极少存在
  - **同义词** : 产生冲突的两个关键码

散列函数选取

$$
Address = Hash(Key)
$$

- 运算简单, 值域在表长范围内, 尽可能单射

常用散列函数

- 除余法 : $hash(key) = key/M$, 其中 M 是一个接近散列长度的质数
  - 缺点 : 连续的关键码映射成连续的散列值, 占据连续数组单元, 可能导致性能降低
- 乘余取整法 : $hash(key) = \lfloor n\{Akey\} \rfloor$, 其中 $A \in (0,1)$
  - 若地址空间为 p, 就取 $n = 2^p$, 则地址恰是小数点后 p 位
  - A 可以取任何值, 与待排序的数据特征有关, 一般取黄金分割最理想
- 平方取中法 : 先求平方, 再取其中几位或其组合作为地址
- 数字分析法 : 对 n 个 d 位数, 每一位可能有 r 种不同的符号, 在各位上出现频率不一定相同. 可根据散列表大小, 选取期中分布均匀的若干位作为地址.
-