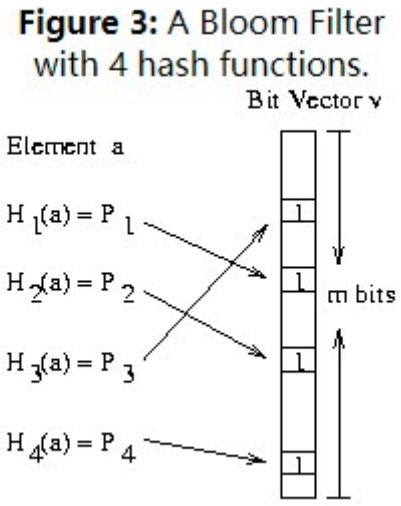
Bloom Filter 实验

简介

Bloom filter 提供了一种快速查找元素是否在集合内的方式，以下简称 BF。BF 利用大小为 m 的哈希数组存储哈希值是否已被插入，取哈希值范围为 [0, m-1] 的 k 个哈希函数（k 个哈希函数可由一个哈希函数通过小变化生成，如第二个哈希函数 $H_2(x)$可以被构造为 $H_1(x + 1)$，对初始的哈希函数进行一个偏移即可）。BF 假设输入元素个数已经确定为 n，每次插入一个元素会计算其 k 个哈希函数的哈希值，并将哈希数组对应的位置置为 1。如下图，BF 的 k 值为 4，对待插入的元素 a 计算了四个哈希值，并将哈希数组的对应位置置为 1。



BF 通过该方式记录元素是否存在集合中会存在一定的误报率。BF 通过规定待插入的元素个数 n、哈希数组的存储大小 m 和哈希函数的个数 k 三个变量可以得到理论上的误报率。推理过程在这里省略，可以参考课件和该链接加深对 BF 的理解。

BF 的误报率最终可由如下公式表示：

$$\left(1-\left(1-\frac{1}{m}\right)^{kn}\right)^{k} \approx \left(1-e^{kn/m}\right)^{k}$$

当 k 取 $k = ln2 \cdot \left(\frac{m}{n}\right)$ 时可以获得理论上最小的误报率。

实验部分

初始条件的选取对实验结果很重要，在实验中选取

1. 待插入的 n 个元素为 0 到 99，也可自行选择输入集合
2. 测试误报率用的测试集合为 100 到 199，也可自行选择测试集合
3. Hash 算法可由自己设计，也可使用 c++提供的 std::hash

控制 m/n 和 k 的值作为变量分别记录多次实验的误报率，示例如下。你只需要完成 m/n 在 2 到 5，k 在 1 到 5 部分的数据即可。

记录完成数据后，请分析你的数据，观察规律并进行分析。同时观察 k 值是否在理论值下误报率最小，如果有差别请分析可能的原因。自行选择输入集合或测试集合的同学，请说明你的思路及输入集合和测试集合。

Table 3: False positive rate under various *m*/*n* and *k* combinations.

m/*n*	*k*	*k*=1	*k*=2	*k*=3	*k*=4	*k*=5	*k*=6	*k*=7	*k*=8
2	1.39	0.393	0.400						
3	2.08	0.283	0.237	0.253					
4	2.77	0.221	0.155	0.147	0.160				
5	3.46	0.181	0.109	0.092	0.092	0.101			

请将你的实验思路、过程和结果写成报告提交，与代码一起做成压缩包上传。