

hw3

姓名：杜心敏

学号：521021910952

1. Hash函数设计

本次实验 Hash 输入为整型数，映射为0—m-1之间的整型数。

bit_m 是Bit数组大小

```
1 int BloomFilter::Hash(const int value) const {  
2     return (value * 17 / 11) % bit_m;  
3 }
```

第i个 Hash 函数，由第一个生成，因为输入是100个，所以每次的间隔选取了107

```
1 Bit [ Hash(key + i * 107) ]
```

2. 测试集

- input: 0—99的int数
- test: 100—100100十万个错误测试集

3. 实验结果

```
1 m = 200  
2 k = 1; faults:49981 0.49981  
3 k = 2; faults:70128 0.70128  
4 k = 3; faults:91411 0.91411  
5 k = 4; faults:96001 0.96001  
6 k = 5; faults:100000 1  
7  
8 m = 300  
9 k = 1; faults:33281 0.33281  
10 k = 2; faults:32633 0.32633  
11 k = 3; faults:31211 0.31211  
12 k = 4; faults:35076 0.35076  
13 k = 5; faults:43365 0.43365  
14  
15 m = 400  
16 k = 1; faults:24982 0.24982  
17 k = 2; faults:24506 0.24506  
18 k = 3; faults:22728 0.22728  
19 k = 4; faults:26300 0.263  
20 k = 5; faults:34397 0.34397
```

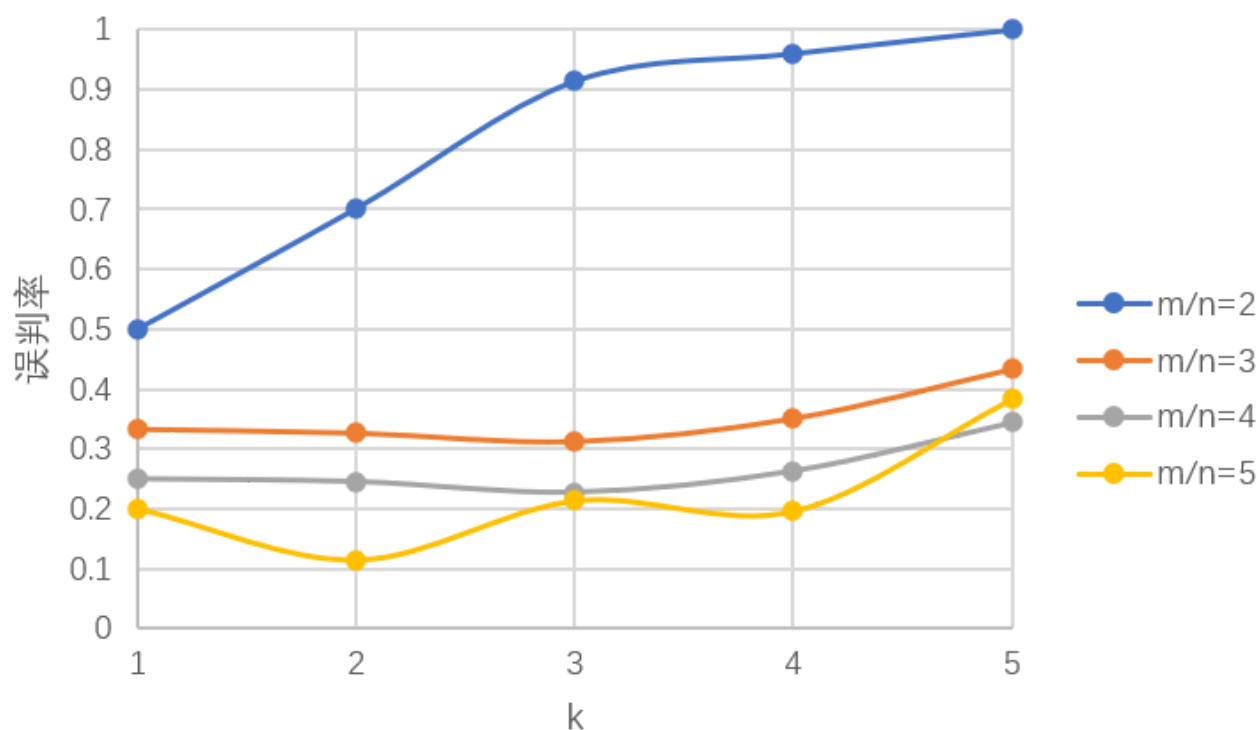
```

21
22 m = 500
23 k = 1; faults:19982 0.19982
24 k = 2; faults:11403 0.11403
25 k = 3; faults:21322 0.21322
26 k = 4; faults:19574 0.19574
27 k = 5; faults:38349 0.38349
28

```

m/n	k理论值	k=1	k=2	k=3	k=4	k=5
2	1.39	0.4998	0.7013	0.9141	0.9600	1.0000
3	2.08	0.3328	0.3263	0.3121	0.3508	0.4337
4	2.77	0.2498	0.2451	0.2273	0.2630	0.3440
5	3.46	0.1998	0.1140	0.2132	0.1957	0.3835

4. 数据分析



- 基本上，最低值落在理论值处
- m/n = 5与理论值偏差1，可能的原因是Hash函数，在Bit数组增大后，并不能很好的将数据均匀映射到Bit数组上。
- 其中，m/n = 2，k = 5的时候出现错误率为1的情况，可能的原因是Bit数组太小，多次hash映射使得数组的值全部被set成1，导致查询表失效。