Homework 3: Bloom Filter实验报告

实验流程:

- 借鉴 *Thomas Wang's 32 bit Mix Function*的方法编写哈希函数，对于不同的函数，$H_i(x)$先进行一次哈希映射，再采用上述方法得到key，将key%m位设置为1，具体哈希映射代码如下：

```
size_t hash(int key, int seed) const {
        key = key << seed + key*seed;
        key = ~key + (key << 15); // key = (key << 15) - key - 1;
        key = key ^ (key >>12);
        key = key + (key << 2);
        key = key ^ (key >>4);
        key = key * 2057; // key = (key + (key << 3)) + (key << 11);
        key = key ^ (key >>16);
        return key;
}
```

- 采用k为15~~，m/n为25~~进行实验，每次插入数范围为1-199，查找200-399是否在其中实验代码如下:

```cpp
int main() {
    int m, k;
    for(int i=1;i<=5;i++){
        std::cout<<"\t"<<i;
    }
    std::cout<<std::endl;
    for(m=2;m<6;m++){
        std::cout<<m<<":\t";
        for(k=1;k<=5;k++){
            BloomFilter filter(m*200, k);
            // 插入n个0-99随机数
            for (int i = 0; i < 200; i++) {
                filter.insert(i);
            }
            // 测试误报率
            int false_positives = 0;
            for (int i =200; i < 400; i++) {
                if (filter.contains(i)) {
                    false_positives++;
                }
            }
            double false_positive_rate = (double)false_positives / 200.00;
            std::cout<< false_positive_rate << "\t";
        }
        std::cout<<std::endl;
    }
    return 0;
}
```

实验结果：

其中，每行为k的值，每列为m的值。

	1	2	3	4	5
2:	0.385	0.405	0.475	0.51	0.55
3:	0.24	0.27	0.305	0.295	0.325
4:	0.2	0.16	0.195	0.18	0.175
5:	0.145	0.115	0.165	0.1	0.17

经计算，当m/n取{2,3,4,5}，k取{1-2，2，2-3，3-4}时，理论上会得到最小报错率，除m=3一组在k=1时得到最小报错率外，其他和理论假设大致相同。

原因可能在于，误报率的公式假设了一些理想的条件，例如哈希函数是均匀随机分布的。但实际上，哈希函数并不总是均匀随机分布的，哈希函数的选择可以影响误报率的计算，不同的哈希函数可能会导致不同的误报率，（如一开始在实验中采用了C++函数的hash库，部分hash函数会导致报错率一直为1或0）这可能会导致误报率的计算与实际情况不符；实验中使用的数据集与理论计算中使用的数据集不同。如果实验中使用的数据集与理论计算中使用的数据集不同，那么误报率也可能不同。