

**2021** 级

《大数据存储与管理》课程

**实 验 报 告**

**姓 名 匡昱恒**

**学 号 U202115337**

**班 号 CS2101**

**日 期 2024.4.22**

[一、选题 3](#_Toc164721018)

[二、实验结构 4](#_Toc164721019)

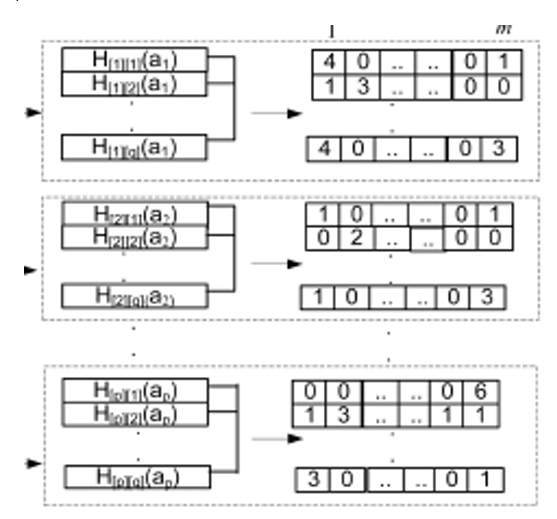
[2.1 数据结构的设计 4](#_Toc164721020)

[2.2 算法的设计 4](#_Toc164721021)

[三、实验总结体会 6](#_Toc164721022)

# 一、选题

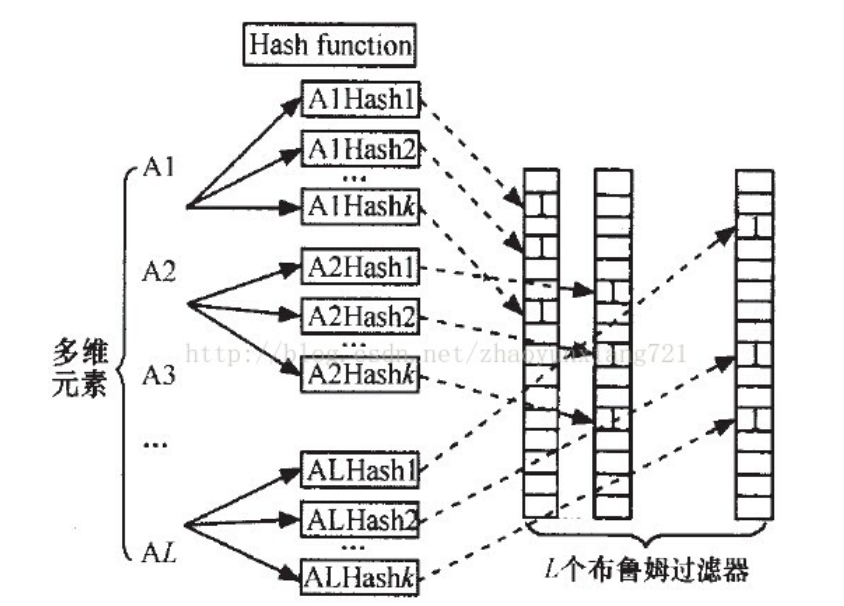
选题1： 基于Bloom Filter的多维数据属性表示和索引



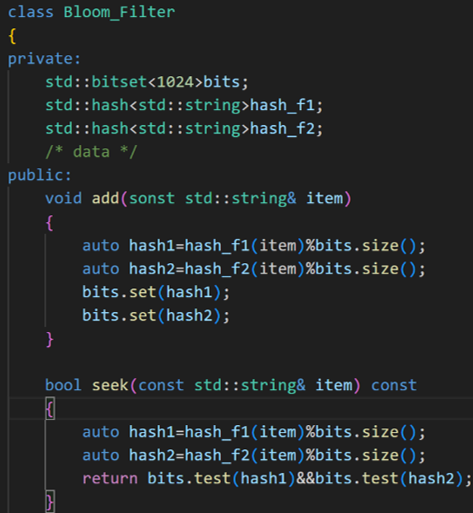
# 二、实验结构

## 2.1 数据结构的设计

在处理多维数据时，采用基于Bloom Filter的数据结构，会用到多维布鲁姆过滤器。多维布鲁姆过滤器采用多个Bloom Filter组成，其个数等于所需存储数据的维数。在元素查询过程中，通过多维元素查询各属性值是否存在相应过滤器中。结构如图所示。



## 2.2 算法的设计

采用C++语言来实现。首先定义Bloom Filter的类。使用C++标准库中的bitset存储位数组，哈希函数用C++标准库的std::hash计算哈希值。为了处理多维数据，需要创建一个包含多个Bloom Filter的结构（多维布鲁姆过滤器），其中每个维度都有一个对应的Bloom Filter。最后再额外使用一个Bloom Filter用于存储属性的联合值。其代码如下图所示：

主函数用来测试，具体代码如下：

int main()

{

MulDimenBloomFilter mdbf;

mdbf.add({ "a","b","c", });

mdbf.add({ "d","e","f", });

std::cout << " TEST 'a','d','b':"

<< (mdbf, seek({ "a","d","b" })) ? "Found" : "Not Found" << std : endl

std::cout << " TEST 'a','d','e':"

<< (mdbf, seek({ "a","d","e" })) ? "Found" : "Not Found" << std : endl

std::cout << " TEST 'a','b','c':"

<< (mdbf, seek({ "a","b","c" })) ? "Found" : "Not Found" << std : endl

}

测试结果为

TEST ‘a’,’d’,’b’: NOT FOUND

TEST ‘a’,’d’,’e’: NOT FOUND

TEST ‘a’,’b’,’c’: FOUND

结果符合预期。

# 三、实验总结体会

本次实验主要完成了对多维联合Bloom Filter的具体代码实现，通过将数据对象的多维属性映射到Bloom Filter中，实现快速的数据定位查询。