

# 实验五 查找表的实现与应用

## 实验目的

1. 掌握查找表的存储结构，实现查找表的基本操作
2. 合理设计运用数据结构，编写有效算法解决 C 代码相似性度量问题

## 实验内容

### 题 1：C 源程序相似性度量问题

为了对比不同 C 语言源程序文件之间的相似性，一种可行的方法是通过扫描给定的不同源程序，统计其中使用 C 语言关键字出现的频度，并利用频度特征向量之间的余弦距离量不同源程序之间的相似程度。

为保证查找效率，现要求使用哈希表统计 C 语言关键字的使用频度。令 C 语言中某长度为 n 的关键字串为 key，则一种可行的哈希函数如下所示：

$$\text{Hash} (\text{key}) = (\text{key}[0]*100+\text{key}[n-1]) \%41$$

其中，表长 m 取 43。请设计采纳有效的冲突处理方法，保证所建哈希表的平均查找长度不大于 2。

## 实验要求

本次实验要求实现如下基本功能：

1. 创建和操作哈希表：根据 `key.txt` 文件提供的 C 语言关键字集合创建哈希表，实现哈希表的查找和插入操作。
2. 输出平均查找长度：根据已经建好的哈希表，假设等概率查找，输出 C 语言关键字的平均查找长度。
3. 计算和判定相似性：对给定的不同源程序文件进行扫描，利用哈希表统计获得 C 语言关键字的使用频度，构成频度特征向量，根据参考示例给出的相似

度计算方法进行判定并输出相似性判定结果。

请认真阅读实验内容，根据实验要求做好问题分析与数据建模，在此基础上完成详细代码设计和上机调试。建议在 6 学时内完成本次实验的所有内容。

## 参考示例

若对程序 1 和程序 2 进行扫描，形成的关键字频度哈希表如下所示：

表 5-1 关键字频度哈希表

关键字	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	...
程序 1 中关键字频度	4	3	0	4	3	0	7	0	0	2	...
程序 2 中关键字频度	4	2	0	5	4	0	5	2	0	1	...
哈希地址	0	1	2	3	4	5	6	7	8	9	...

其中，根据程序 1 和程序 2 中关键字出现的频度，可提取到两个程序的特征向量  $X_1$  和  $X_2$ ，其中：

$$X_1 = (4 \ 3 \ 0 \ 4 \ 3 \ 0 \ 7 \ 0 \ 0 \ 2 \ \dots)^T$$
$$X_2 = (4 \ 2 \ 0 \ 5 \ 4 \ 0 \ 5 \ 2 \ 0 \ 1 \ \dots)^T$$

一般情况下，可以通过计算向量  $X_1$  和  $X_2$  的余弦相似度来判断对应两个程序的相似性，余弦相似度的计算公式为：

$$S(X_i, X_j) = \frac{X_i^T \cdot X_j}{|X_i| \cdot |X_j|} \quad (5-1)$$

其中  $|X_i| = \sqrt{X_i^T \cdot X_i}$ 。 $S(X_i, X_j)$  的值介于 [0,1] 之间，也称广义余弦，即  $S(X_i, X_j) = \cos\theta$ 。

$X_i = X_j$  时，显见  $S(X_i, X_j) = 1, \theta = 0$ ； $X_i$ 、 $X_j$  差别很大时， $S(X_i, X_j)$  接近 0， $\theta$  接近  $\pi/2$ 。如  $X_1 = (1 \ 0)^T$ ,  $X_2 = (0 \ 1)^T$ ，则  $S(X_i, X_j) = 0$ ,  $\theta = \pi/2$ 。可以用下图来直观地展示向量的相似程度。

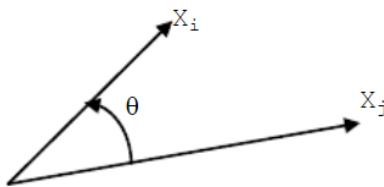


图 5-1 两个特征向量之间的余弦相似度

以下给出了 32 个 C 语言关键字的一种哈希存储模式及针对不同源程序的词频统计情况：

哈希序号	关键字	频数 1	频数 2	频数 3	频数 4
0	enum	0	0	0	0
1	switch	0	0	0	0
2	union	0	0	0	0
3	case	0	0	0	0
4	extern	0	0	0	0
5	sizeof	0	0	0	0
10	char	0	1	0	1
11	void	13	16	11	10
12	auto	0	0	0	0
13	short	2	2	0	0
14	struct	1	3	13	5
15	double	19	19	25	0
16	const	2	2	5	0
17	typedef	1	3	1	4
18	volatile	0	0	0	0
23	for	16	18	13	0
24	if	9	9	15	8
25	float	3	3	0	0
26	do	0	0	0	0
27	break	2	2	4	1
29	while	0	0	2	6
30	default	0	0	0	0
31	return	1	3	9	1
33	unsigned	4	4	1	0
34	else	2	2	10	7
35	register	0	0	0	0
37	static	0	0	0	0
38	int	18	22	27	8
39	long	0	0	0	0
40	signed	0	0	0	0
41	continue	0	0	0	0
42	goto	1	1	0	0

图 5-2 关键字的哈希存储及词频统计示例

设定  $S(X_i, X_j)$  判定阈值为 0.85，以下给出了几个 C 源程序之间的相似度比较

情况：

```
test\test1.cpp 和 test\test2.cpp 的相似情况为：相似度 xs=0.991
这两个文件内容相似
test\test1.cpp 和 test\test3.cpp 的相似情况为：相似度 xs=0.886
这两个文件内容相似
test\test1.cpp 和 test\test4.cpp 的相似情况为：相似度 xs=0.459
这两个文件内容不相似
```

图 5-3 几种不同 C 源程序的相似度输出结果示例