

Project 7 哈希表设计

21307035 邓栩瀛

1、程序功能简要说明。

- 针对所在的班级中的“姓名”设计了一个哈希表，平均查找长度不超过L，具有相应的建表和查表的功能。
- 哈希函数采用除留余数法的方式进行构造，并用伪随机探测再散列法处理冲突。
- 姓名是汉语拼音的形式，填入哈希表的姓名共有30个，平均查找长度的上限为2。

2、程序运行截图，包括计算功能演示、部分实际运行结果展示、命令行或交互式界面效果等。

- 输出界面

```
Enter 1:显示哈希表
Enter 2:查找
Enter 0:Exit
Please enter your choice!
```

- 哈希表展示（包括地址、关键字、搜索长度、 $H(key)$ 、姓名、平均查找长度）

```
Enter 1:显示哈希表
Enter 2:查找
Enter 0:Exit
Please enter your choice!
1
地址      关键字      搜索长度      H(key)      姓名
0         1150        1            10          Huang Lan Shu
1         1251        1            21          Wang Liang Zhi
2         702         1            12          Pang Hao
3         0           0            0
4         954         1            24          Lin Wen Jun
5         1055        1            5           Wei Jia Zhuo
6         1256        1            26          Liang Zhi Zhao
7         0           0            0
8         1058        1            8           Deng Xu Ying
9         1159        1            19          Li Zhong Tian
10        0           0            0
11        0           0            0
12        1512        1            12          Zhong Zheng Zhou
13        943         6            13          Wen Zi Jian
14        0           0            0
15        1165        1            25          Liu Kun Sheng
16        0           0            0
17        608         2            8           Yang Bo
18        0           0            0
19        819         1            9           Li Hao Hui
20        0           0            0
21        0           0            0
22        0           0            0
23        0           0            0
24        0           0            0
25        0           0            0
26        1226        1            26          Zheng Bei Chen
27        727         1            7           Jin Si Qi
28        0           0            0
29        819         2            9           Ma Jun Jie
30        0           0            0
```

31	1231	1	1	Chen Jin Qiang
32	832	1	22	Liu Jia Yi
33	926	2	26	Ma Feng Rui
34	834	1	24	Huang Xin
35	0	0	0	
36	936	1	6	Chen Rui Qi
37	1037	1	17	He Qian Ying
38	0	0	0	
39	0	0	0	
40	0	0	0	
41	1091	1	11	Zhong Xin Yu
42	942	1	12	Sun Hang Qi
43	1143	1	3	Zhang Wen Kai
44	1344	1	24	Liang Quan Feng
45	1145	1	5	Zhang Jia Jun
46	0	0	0	
47	847	1	7	Zheng Yue
48	1248	1	18	Liu Ming Liang
49	1049	1	29	Xu Zhan Chao

平均查找长度: ASL(30)=1.266667

- 键入某字符串进行搜索，查找成功时，显示姓名及关键字，并计算和输出查找长度；查找失败则显示"Not Found!"

```
Enter 1:显示哈希表
Enter 2:查找
Enter 0:Exit
Please enter your choice!
2
请输入姓名的拼音:Deng Xu Ying
姓名:Deng Xu Ying 关键字:1058 查找长度: 1
```

```
Enter 1:显示哈希表
Enter 2:查找
Enter 0:Exit
Please enter your choice!
2
请输入姓名的拼音:Zhang San
Not Found!
```

3、部分关键代码及其说明。

- 哈希表结构

```
1 typedef struct HASH           //哈希表
2 {
3     char const* name;         //名字的拼音
4     int key;                  //拼音所对应的关键字
5     int size;                 //查找长度
6 }HASH;
```

- 哈希表的初始化及测试数据集

```
1 void InitNameList()
2 {
3     int r,i;
4     NameList[0].name = "Huang xin";
5     NameList[1].name = "Zheng Bei Chen";
6     NameList[2].name = "Xu Zhan Chao";
7     NameList[3].name = "Huang Lan Shu";
8     NameList[4].name = "Lin Wen Jun";
9     NameList[5].name = "Deng Xu Ying";
```

```

10    NameList[6].name = "Zhong Xin Yu";
11    NameList[7].name = "Zhang Jia Jun";
12    NameList[8].name = "Jin Si Qi";
13    NameList[9].name = "Chen Jin Qiang";
14    NameList[10].name = "Zhong Zheng Zhou";
15    NameList[11].name = "Li Zhong Tian";
16    NameList[12].name = "He Qian Ying";
17    NameList[13].name = "Li Hao Hui";
18    NameList[14].name = "Zheng Yue";
19    NameList[15].name = "Chen Rui Qi";
20    NameList[16].name = "Liu Jia Yi";
21    NameList[17].name = "Wei Jia Zhuo";
22    NameList[18].name = "Ma Jun Jie";
23    NameList[19].name = "Liu Ming Liang";
24    NameList[20].name = "Liu kun Sheng";
25    NameList[21].name = "Wang Liang Zhi";
26    NameList[22].name = "Sun Hang Qi";
27    NameList[23].name = "Liang Zhi Zhao";
28    NameList[24].name = "Liang Quan Feng";
29    NameList[25].name = "Zhang Wen Kai";
30    NameList[26].name = "Wen Zi Jian";
31    NameList[27].name = "Ma Feng Rui";
32    NameList[28].name = "Pang Hao";
33    NameList[29].name = "Yang Bo";
34    for (i = 0; i < NAME_NO; i++)
35    {
36        for (r = 0; (*(NameList[i].name + r)) != '\0'; r++)
37            NameList[i].key += toascii(*(NameList[i].name+r)); //运用
38        toascii函数
39    }

```

- 创建哈希表（哈希函数采用除留余数法的方式进行构造，并用伪随机探测再散列法处理冲突。）

```

1  void CreateHashList()
2  {
3      int i;
4      for (i = 0; i < HASH_LENGTH; i++)
5      {
6          HashList[i].name = "";
7          HashList[i].key = 0;
8          HashList[i].size = 0;
9      }
10     for (i = 0; i < NAME_NO; i++)
11     {
12         int sum = 0;
13         int adr = (NameList[i].key) % M; //哈希函数
14         int d = adr;
15         if (HashList[adr].size == 0) //如果不冲突
16         {
17             HashList[adr].key = NameList[i].key;
18             HashList[adr].name = NameList[i].name;
19             HashList[adr].size = 1;
20         }
21         else //冲突

```

```

22         {
23             do
24             {
25                 d = (d + NameList[i].key % 10 + 1) % M; //伪随机探测再散列法处
理冲突
26                 sum = sum + 1; //查找次数加1
27             } while (HashList[d].key != 0);
28             HashList[d].key = NameList[i].key;
29             HashList[d].name = NameList[i].name;
30             HashList[d].size = sum + 1;
31         }
32     }
33 }

```

- 查找函数的实现

```

1 void FindList() //查找
2 {
3     char name[20];
4     int s0 = 0, r, sum = 1, adr, d;
5     printf("请输入姓名的拼音:");
6     gets(name);
7     for (r = 0; name[r] != '\0'; r++) //求出姓名的拼音所对应的整数(关键字)
8         s0 += name[r];
9     adr = s0 % M; //使用哈希函数
10    d = adr;
11    if (HashList[adr].key == s0) //分3种情况进行判断
12        printf("\n姓名:%s 关键字:%d 查找长度: 1", HashList[d].name, s0);
13    else if (HashList[adr].key == 0)
14    {
15        printf("Not Found!");
16    }
17    else
18    {
19        int g = 0;
20        do
21        {
22            d = (d + s0 % 10 + 1) % M; //伪随机探测再散列法处理冲突
23
24            sum = sum + 1;
25            if (HashList[d].key == 0)
26            {
27                printf("Not Found!");
28                g = 1;
29            }
30            if (HashList[d].key == s0)
31            {
32                printf("\n姓名:%s 关键字:%d 查找长度:%d",
HashList[d].name, s0, sum);
33                g = 1;
34            }
35        } while (g == 0);
36    }
37 }

```

4、程序运行方式简要说明。

代码内已内置30个姓名填入哈希表，可以输出构造的哈希表；通过键入任意字符串，输出查找结果。