

实验 4-哈希表

PB17111614

王嵘晟

1. 实验要求

掌握哈希表的定义和特点

掌握哈希函数的构造方法和解决冲突的技术，实现哈希造表

掌握哈希表的查找技术

掌握哈希表平均查找长度 ASL 的计算方法

2. 实验内容（大体简述实验过程）

用除留余数法构造 Hash 函数，分别用两种方法处理冲突：线性探测法和链地址法。

用 C 语言作为编译语言，输入数据个数 n 表长 m 模值 p

输出两个哈希表以及两种方式分别的查找成功与查找失败平均查找长度。

再次输入要查找的元素，查找成功输出 Success，失败输出 Unsuccess，并输出查找长度。

3. 实验关键代码讲述（可结合图片、文字说明）

```
for(i=0;i<n;i++)
{
    H1[i]=H[i];
    H2[i]=H[i];
}
memset(HT1,0,sizeof(HT1));
memset(HT2,0,sizeof(HT2));
int count1=n,count2=n;
int count3=0,count4=1;
for(k=0;k<m;k++)
{
    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(H1[i]==H1[j])
            {
                H1[j]+=1;
                H1[j]%=m;
                count1++;
            }
        }
    }
}
```

将 Hash 函数值分为两份分别用两种方法处理冲突，使用线性探测处理冲突并记查找成功的平均查找长度

7	
3	
3	—
3	
1	—
2	
3	
4	—
5	
5	

```

printf("\nH1(key):\n");
for(i=0;i<n;i++)
{
    printf("%d  ",H1[i]);
}
printf("\n\n");
for(i=0;i<n;i++)
{
    HT1[H1[i]]=key[i];
}

```

构造 Hash 表 1

```

for(i=0;i<m;i++)
{
    if(HT1[i]==0)
    {
        count3++;
        continue;
    }
    for(j=0;j<m;j++)
    {
        if(HT1[j]==0 && j>=i)
        {
            count3+=j-i+1;
            break;
        }
        else
        {
            for(k=i+1;k<m;k++)
            {
                if(HT1[k]==0)
                {
                    break;
                }
            }
            if(k==m)
            {
                if(HT1[j]==0 && j<i)
                {
                    count3+=m+1-i+j;
                    break;
                }
            }
        }
    }
}
}

```

表 1 中查找失败的平均查找长度

```

for(i=0;i<n-1;i++)
{
    for(j=i+1;j<n;j++)
    {
        if(H2[i]>H2[j])
        {
            tmp=H2[j];
            H2[j]=H2[i];
            H2[i]=tmp;
            tmp=key[j];
            key[j]=key[i];
            key[i]=tmp;
        }
    }
}
for(i=0;i<n-1;i++)
{
    for(j=i+1;j<n;j++)
    {
        if(H2[i]==H2[j])
        {
            if(key[i]>key[j])
            {
                tmp=key[j];
                key[j]=key[i];
                key[i]=tmp;
            }
        }
    }
}

```

对 key 根据 Hash 函数值由小到大排序，并根据 key 大小由小到大排序

```

j=0;
for(i=0;i<n;i++)
{
    if(H2[i]>H2[i-1])
        j=0;
    for(k=0;k<m;k++)
    {
        if(k==H2[i])
        {
            HT2[k][j++]=key[i];
        }
    }
}

```

构建 Hash 表 2

```

for(i=0;i<m;i++)
{
    for(j=0;j<m;j++)
    {
        if(HT2[i][j]!=0)
        {
            count2+=j;
            count4++;
        }
    }
}

```

表 2 中 ASL，成功和失败分别为 count2 和 count4

```
int aim,temp,count5=0,count6=0;
int flag=0;
printf("\nPlease enter the number you want to search for:\n");
scanf("%d",&aim);
temp=aim%p;
for(i=0;i<n;i++)
{
    if(key[i]==aim)
    {
        if(H1[i]>=H[i])
            count5=H1[i]-H[i]+1;
        else
            count5=m-H1[i]+H[i]+1;
        flag=1;
        break;
    }
}
```

输入元素进行查找，表 1 中查找成功

```
if(count5==0)
{
    for(i=temp+1;i<m;i++)
    {
        if(HT1[i]==0)
        {
            count5=i-temp+1;
            break;
        }
    }
    if(i==m)
    {
        for(j=0;j<m;j++)
        {
            if(HT1[j]==0)
            {
                count5=m-temp+j+1;
                break;
            }
        }
    }
}
```

表 1 中查找失败

```
for(i=0;i<m;i++)
{
    if(HT2[temp][i]==aim)
        count6=i+1;
}
if(count6==0)
{
    for(i=0;i<m;i++)
    {
        if(HT2[temp][0]==0)
        {
            count6++;
            break;
        }
        if(HT2[temp][i]!=0)
            count6++;
    }
}
```

表 2 中查找成功与失败

4. 实验结果及分析（结合相关数据截图分析）

```
Please enter the number of the keys:
12
Please enter the length of Hash table:
13
Please enter the divider:
13
Please enter 12 keys:
19 14 23 1 68 20 84 27 55 11 10 79

H(key):
6   1   10  1   3   7   6   1   3   11  10  1

H1(key):
6   1   10  2   3   7   8   4   5   11  12  9

HT1:
0   14   1   68   27   55   19   20   84   79   23   11   10

Success:
ALS1=30/12

Unsuccess:
ALS1'=91/13
```

```
H2[key]:
1   1   1   1   3   3   6   6   7   10  10  11
HT2:
0   ^
1   →19→14→23→1^
2   ^
3   →68→20^
4   ^
5   ^
6   →84→27^
7   →55^
8   ^
9   ^
10  →11→10^
11  →79^
12  ^

Success:
ASL2=21/12

Unsuccess:
ASL2'=25/13
```

```
Please enter the number you want to search for:
99

Unsuccess:
Table 1:  6
Table 2:  1
```

查找的 99 是表里没有的

```
Please enter the number you want to search for:
84

Success:
Table 1:  3
Table 2:  1
```

查找的 84 在表里，给出查找成功的查找长度

5. 实验小结

通过本次试验，学习到了 Hash 表的实际应用。学习到了如何构建 Hash 表，如何算 ASL 还有如何利用 Hash 表来查找元素。