

中山大学计算机学院本科生实验报告

(2021 学年第 1 学期)

课程名称: Data structures and algorithms

任课教师: 张子臻

年级	20 级	专业 (方向)	软件工程
学号	20337270	姓名	钟海财
电话	13397996670	Email	2940599563@qq.com
开始日期	2021/11/10	完成日期	2021/11/28

1. 实验题目

Complete a small text searching engine using hash tables.

2. 实验目的

1. 使用哈希表完成一个小的文本搜索引擎, 通过输入一个单词, 查询文本中所有包含该单词的句子并输出。
2. 尝试使用不同的哈希函数和冲突解决方法。

3. 程序设计

3.1 哈希函数: 共 6 种, 包含在 myhash.cpp 里

```
1. #include<iostream>
2. #include<string>
3. using namespace std;
4. int myhash1(const string s,int n){//BKDRHash 算法
5.     int hash = 0 ;
6.     int i = 0 ;
7.     while(s[i]){
8.         hash =(hash*131) + (int)(s[i++]);
9.     }
10.    return (hash & 0x7FFFFFFF)%n;
11. }
```

```

12.
13. int myhash2(const string str,int n){//DJBHash 算法
14.     int hash = 5381;
15.     int i = 0;
16.     while(str[i])
17.         hash += (hash << 5) + (str[i++]);
18.     return (hash & 0x7FFFFFFF)%n;
19. }
20.
21. int myhash3(const string str,int n){//JSHash 算法
22.     int hash = 1315423911;
23.     int i = 0;
24.     while(str[i])
25.         hash ^= ((hash << 5) + (str[i++]) + (hash >> 2));;
26.     return (hash & 0x7FFFFFFF)%n;
27. }
28.
29. int myhash4(const string str,int n){//RSHash 算法
30.     unsigned int b = 378551;
31.     unsigned int a = 63689;
32.     unsigned int hash = 0;
33.     int i = 0;
34.     while (str[i]){
35.         hash = hash * a + (str[i++]);
36.         a *= b;
37.     }
38.     return (hash & 0x7FFFFFFF)%n;
39. }
40.
41. int myhash5(const string str,int n){//SDBMHash 算法
42.     unsigned int hash = 0;
43.     int i = 0;
44.     while (str[i]){
45.         // equivalent to: hash = 65599*hash + (*str++);
46.         hash = (str[i++]) + (hash << 6) + (hash << 16) - hash;
47.     }
48.     return (hash & 0x7FFFFFFF)%n;
49. }
50.
51. int myhash6(const string str,int n){//ELFHash 算法
52.     unsigned int hash = 0;
53.     unsigned int x = 0;
54.     int i = 0 ;
55.     while (str[i]){
56.         hash = (hash << 4) + (str[i++]);
57.         if ((x = hash & 0xF000000L) != 0){

```

```

58.             hash ^= (x >> 24);
59.             hash &= ~x;
60.         }
61.     }
62.     return (hash & 0x7FFFFFFF)%n;
63. }

```

3.2 判断是否为单词的函数，提取字符串里的单词的函数

```

1. bool isword(char c){ //判断是不是组成单词的数字或字母
2.     if((int)c>=48 && (int)c <=57) return true;
3.     else if((int)c>=65 && (int)c <=90) return true;
4.     else if((int)c>=97 && (int)c <=122) return true;
5.     else return false;
6. }
7. void getword(string s,vector<string>& ss){
8.     //提取字符串 s 中的所有单词到 vector<string>&ss 中
9.     string word = "";
10.    int k = 0 ;
11.    while(k<s.size()){
12.        word = "";
13.        while(k<s.size()&& !isword(s[k])){
14.            ++k;
15.        }
16.        while(k<s.size()&& isword(s[k])){
17.            word.push_back(s[k]);
18.            ++k;
19.        }
20.        if(word!="") ss.push_back(word);
21.    }
22. }

```

3.3 开放地址法函数设计

3.3.1 读取 text.txt 文件并处理成哈希表的函数 _read

```
bool _read(vector<vector<string> >&all,vector<vector<int> > &table,string* hash_table, int n)
```

其中 n 为 hash_table 的容量；vector<vector<string> >&all 用于存储 text.txt 里的句子,all[k]表示第 k 个句子；hash_table[]为存放单词的哈希表；table 用于储存对应单词存在的句子序号，如该单词在哈希表中位置为 hash_table[key]时，

存在的句子的序号保存在 `table[key][]` 中。

每次从 `text.txt` 里读入一个句子时(以 '!' , '!', '?' 结尾), 都将该句子 `push_back` 进 `all` 里面。再对这个句子进行处理, 将该句子中的每个单词都存到哈希表 `hash_table` 里面, 对应的哈希值由 `getkey` 函数生成, 使用开放地址法里的平方探测法解决冲突:

```
1. int getkey(string table[],string s,int n){
2. //生成单词 s 在哈希表中已经解决冲突的 key 值
3.     ++all_words;
4.     int key = myhash(s,n); //用哈希函数生成哈希值
5.     int k = 0;
6.     ++search;
7.     while(table[(key+k*k)%n] != "" ){ //平方探测法解决冲突
8.         if(table[(key+k*k)%n] == s ) return (key+k*k)%n;
9. //如果该单词已经在哈希表 table[] 中时返回该 key 值
10.         ++k;
11.         ++search;
12.     }
13.     ++diff_words;
14.     return (key+k*k)%n; //返回已经解决冲突的 key 值
15. }
```

再将该句子序号 `i` 存到 `table` 中的对应位置, 例如某单词 `s` 通过 `getkey` 得到的 `key=getkey(hash_table,s,n)`, 在 `table` 中对应位置为 `table[key]`, 如果 `table[key].size()==0` 或 `table[key][table[key].size()-1]!=i`, 说明该句子未被包含在 `table[key][]` 中, 便 `table[key].push_back(i)`, 实现将该句子序号记入对应的 `table` 中并去重。

```
1. bool _read(vector<vector<string> >&all,vector<vector<int> > &table,string* hash_t
   able, int n){
2. //将 text.txt 读到 vector<vector<string> >&all 里,all[k]表示第 k 个句子
3.     ifstream infile("text.txt",ios::in);
4.     if(!infile.is_open()) return false ;
5.     int i = 0;//表示第 i 个句子, 存到 all[i]里
6.     while(!infile.eof()){
7.         vector<string> ss;
8.         string str;
9.         str.clear();
10.         infile>>str;
```

```

11.         while(str[str.size()-1]!='\n' &&str[str.size()-1]!='.' &&!infile.eof() &&
           str[str.size()-1]!='!'&&str[str.size()-1]!='?'){
12.             if(str[str.size()-1]=='"' &&(str[str.size()-2]=='.' || str[str.size()-2]=
           ='!' || str[str.size()-2]=='?')) break;
13.             ss.push_back(str);
14.             infile>>str;
15.         }
16.         ss.push_back(str);
17.         all.push_back(ss);
18.         for(int j = 0; j <all[i].size() ;++j){
19.             vector<string> ss;
20.             getword(all[i][j],ss);//将字符串 all[i][j]里的所有单词提取到 ss 里
21.             for(int k = 0; k <ss.size();++k){
22.                 int key = getkey(hash_table,ss[k],n);
23.                 //生成单词 ss[k]的解决冲突之后的哈希值
24.                 hash_table[key] = ss[k] ; //将单词存到哈希表 hash_table 里
25.                 int h = table[key].size();
26.                 if(h==0 || table[key][h-1]!=i)
27.                     table[key].push_back(i);//将该句子序号记入 table 对应的位置中
28.             }
29.         }
30.         ++i;
31.     }
32.     infile.close();
33.     return true;
34. }
35.

```

3.3.2 开放地址法的主函数设计

主程序定义了三个存储变量：

`vector<vector<string>> all`: 将所有句子读到 `vector<vector<string>>` 类型的 `all` 里面,`all[k]`表示第 `k` 个句子。

`string hash_table[n]`: `hash_table` 为储存单词的哈希表,由 `getkey(string table[],string s,int n)`函数生成对应单词 `s` 解决冲突之后的 `key` 值。

`vector<vector<int>> table`: 用于储存单词所存在的所有句子的序号, 如 `table[key][]`存放了 `hash` 值为 `key` 的单词所在所有句子的序号。

打开输出文件 answer1.txt，将控制台输出同步到 answer1.txt 里。

```
1. int main(){
2.     cout<<"开放地址法: "<<endl;
3.     vector<vector<string> > all;
4.     //将所有句子读到 vector<vector<string> >类型的 all 里面,all[k]表示第 k 个句子
5.     int n = 4900 ;//hash_table 的容量 , 要超过不同单词的个数
6.     cout<<"请输入哈希表的容量 n(推荐 n>=4900): "<<endl ;
7.     cin>>n;
8.     string hash_table[n];
9.     //hash_table 为储存单词的哈希表,由 getKey(string table[],string s,int n)函数生成对应单词 s 解决冲突之后的 key 值
10.    vector<vector<int> > table;
11.    //用于储存单词所存在的所有句子的序号,如 table[key][]存放了 hash 值为 key 的单词所在所有句子的序号
12.    vector<int> t;
13.    for(int i = 0 ; i <n; ++i) table.push_back(t); //对 table 进行初始化赋值
14.    hash_k = 1;//默认使用 BKDRHash 算法的哈希函数 myhash1
15.    cout<<"哈希函数选择: "<<endl
16.        <<"1: BKDRHash 算法"<<endl
17.        <<"2: DJBHash 算法"<<endl
18.        <<"3: JSHash 算法"<<endl
19.        <<"4: RSHash 算法"<<endl
20.        <<"5: SDBMHash 算法"<<endl
21.        <<"6: ELFHash 算法" <<endl;
22.    cout<<"请输入整数 1~6 来选择哈希函数 hash_k : "<<endl;
23.    cin>>hash_k; //输入选择的哈希函数
24.    //将所有句子读到 vector<vector<string> >类型的 all 里面,all[k]表示第 k 个句子
25.    bool A1 = _read(all,table,hash_table,n);//读取 text.txt 文件并生成对应的 all,table 和 hash_table
26.    ofstream outfile("answer1.txt",ios::out);//输出到文件 answer.txt 里
27.    cout<<"all sentences: "<<all.size()<<endl;
28.    cout<<"all_words: "<<all_words<<endl;
29.    cout<<"diff_words: "<<diff_words<<endl;
30.    cout<<"search: "<<search<<endl;
31.    cout<<"space = n : "<<n<<endl;
32.    cout<<"n/diff_words: "<<(double)n/((double)diff_words<<endl;
33.    cout<<"average_search = search/diff_words: "<< (double)search/((double)diff_words <<endl;
34.    cout<<"-----"<<endl;
35.    cout<<"You can use '$stop' to stop the Query !"<<endl;
36.    cout <<"Query : "<<endl; //测试样例 and 和 amu 在没有解决冲突时 hash 值相同
37.    cout<<"请输入: ";
38.    outfile<<"开放地址法: "<<endl;
39.    outfile<<"请输入哈希表的容量 n(推荐 n>=4900): "<<endl<<n<<endl ;
40.    outfile<<"哈希函数选择: "<<endl
```

```

41.         <<"1: BKDRHash 算法"<<endl
42.         <<"2: DJBHash 算法"<<endl
43.         <<"3: JSHash 算法"<<endl
44.         <<"4: RSHash 算法"<<endl
45.         <<"5: SDBMHash 算法"<<endl
46.         <<"6: ELFHash 算法" <<endl;
47.     outfile<<"请输入整数 1~6 来选择哈希函数 hash_k : "<<endl<<hash_k<<endl;
48.     outfile<<"all sentences: "<<all.size()<<endl;
49.     outfile<<"all_words: "<<all_words<<endl;
50.     outfile<<"diff_words: "<<diff_words<<endl;
51.     outfile<<"search: "<<search<<endl;
52.     outfile<<"space = n : "<<n<<endl;
53.     outfile<<"n/diff_words: "<<(double)n/(double)diff_words<<endl;
54.     outfile<<"average_search = search/diff_words: "<< (double)search/(double)diff
    _words <<endl;
55.     outfile<<"-----"<<endl;
56.     outfile<<"You can use '$stop' to stop the Query !"<<endl;
57.     outfile<<"Query :"<<endl;
58.     outfile<<"请输入: ";
59.     string find;//要查询的单词
60.     while(cin>>find && find!="$stop"){ //根据输入的单词进行相应操作如查询或退出
61.         outfile<<find<<endl;
62.         int key = getkey(hash_table,find,n);
63.         if(table[key].size()==0){//该单词 find 不存在对应的句子时
64.             cout<<"There is not the word!"<<endl;
65.             outfile<<"There is not the word!"<<endl;
66.         }
67.         for(int i = 0 ; i <table[key].size();++i){
68.             cout<<"Sentence "<<table[key][i] <<endl;//输出句子序号, 从 0 开始
69.             outfile<<"Sentence "<<table[key][i] <<endl;
70.             for(int j = 0; j <all[table[key][i]].size() ;++j){
71.                 //输出序号为 table[key][i] 的句子
72.                 cout<<all[table[key][i]][j]<<" ";
73.                 outfile<<all[table[key][i]][j]<<" ";
74.             }
75.             cout<<endl;
76.             outfile<<endl;
77.         }
78.         cout<<endl<<"-----"<<endl;
79.         //进行下一次 Query
80.         cout<<"You can use '$stop' to stop the Query !"<<endl;
81.         cout<< "Query :"<<endl;
82.         cout<<"请输入: ";
83.         outfile<<endl<<"-----"<<endl;
84.         outfile<<"You can use '$stop' to stop the Query !"<<endl;
85.         outfile<< "Query :"<<endl;

```

```

86.         outfile<<"请输入: ";
87.     }
88.     outfile.close();
89.     return 0;
90. }

```

3.4 拉链法函数设计

3.4.1 读取 text.txt 文件并处理成哈希表的函数 _read

```
1. bool _read(vector<vector<string> >&all,vector<vector<word*> > &table, int n);
```

这里定义结构体 word:

```

1. struct word{
2.     string s;
3.     vector<int> index;//储存单词 s 存在的句子的序号
4. };

```

all 和开放地址法一样设计。 n 为哈希表 table 的容量。

vector<vector<word*> > table: 储存单词所在句子序号, 如单词 s, key =

getkey(table,s,n),所在句子序号储存在 table[myhash[s]][key]->index[]里。

_read 的读取方式和开放地址法 3.3.1 相似, 每读入一个句子存到 all 里, 就将该句子中的单词存到哈希表 table 里, 使用拉链法解决冲突, 如单词 s 的哈希值为 myhash(s),getkey 函数使用拉链法解决冲突返回 s 在 table[myhash(s)][]的位置。所以 **getkey 函数**:

```

1. int getkey(vector<vector<word*> > &table,string ss,int n){//生成单词 s 在哈希表
    table[key1]拉链中的索引
2.     ++all_words;
3.     int key1 = myhash(ss,n);
4.     int h= table[key1].size();
5.     for(int i = 0 ; i < h;++i){ //拉链法
6.         if(table[key1][i]->s == ss) return i;//如果该单词已经在哈希表 table[]中时返回
            该 key 值
7.     }
8.     search+=(h+1);
9.     ++diff_words;
10.    return table[key1].size();
11. }

```


_read 函数:

```
1. bool _read(vector<vector<string> >&all,vector<vector<word*> > &table, int n){
2. //将 text.txt 读到 vector<vector<string> >&all 里,all[k]表示第 k 个句子
3.     ifstream infile("text.txt",ios::in);
4.     if(!infile.is_open()) return false ;
5.     int i = 0;
6.     while(!infile.eof()){
7.         vector<string> ss;
8.         string str;
9.         str.clear();
10.        infile>>str;
11.        while(str[str.size()-1]!='\n' &&str[str.size()-1]!='.' &&!infile.eof() &&
            str[str.size()-1]!='!'&&str[str.size()-1]!='?'){
12.            if(str[str.size()-1]=='"' &&(str[str.size()-2]=='.' || str[str.size()-2]=
                ='!' || str[str.size()-2]=='?')) break;
13.            ss.push_back(str);
14.            infile>>str;
15.        }
16.        ss.push_back(str);
17.        all.push_back(ss);
18.        for(int j = 0; j <all[i].size() ;++j){
19. //对第 i 个句子 all[i]提取所有单词并存到哈希表里
20.            vector<string> ss;
21.            getword(all[i][j],ss);//将字符串 all[i][j]里的所有单词提取到 ss 里
22.            for(int k = 0; k <ss.size();++k){
23.                int key1 = myhash(ss[k],n);
24.                int key2 = getkey(table,ss[k],n);
25.                //生成单词在拉链中的索引
26.                if(key2==table[key1].size()){ //此时表示单词 s 未在拉链中
27.                    word *t0 = new word;
28.                    t0->s = ss[k] ;
29.                    table[key1].push_back(t0);
30.                }
31.                int h = table[key1][key2]->index.size();
32.                if(h==0 || table[key1][key2]->index[h-1]!=i) //如果该句子未被记录
33.                    table[key1][key2]->index.push_back(i);//将该句子序号记入对应的
                    table 中
34.            }
35.        }
36.        ++i;
37.    }
38.    infile.close();
39.    return true;
40. }
```

3.4.2 拉链法的主函数设计

和开放地址法的主函数设计 3.4.1 类似：

```
1. int main(){
2.     vector<vector<string> > all;
3. //将所有句子读到 vector<vector<string> >类型的 all 里面,all[k]表示第 k 个句子
4.     cout<<"拉链法: " <<endl;
5.     int n = 5000 ;//table 的容量 , 由于使用拉链法, 可以小于不同单词的个数
6.     cout<<"请输入哈希表的容量 n:(推荐 n>=100) " <<endl ;
7.     cin>>n;
8.     vector<vector<word*> > table;
9. //储存单词所在句子序号, 如单词 s, key = getKey(table,s,n),
10. //所在句子序号储存在 table[myhash[s]][key]->index[]里
11.     vector<word*> t;
12.     for(int i = 0 ; i <n; ++i) table.push_back(t); //对 table 进行初始化赋值
13.     hash_k = 1;//默认使用 BKDRHash 算法的哈希函数 myhash1
14.
15.     cout<<"哈希函数选择: " <<endl
16.         <<"1: BKDRHash 算法" <<endl
17.         <<"2: DJBHash 算法" <<endl
18.         <<"3: JSHash 算法" <<endl
19.         <<"4: RSHash 算法" <<endl
20.         <<"5: SDBMHash 算法" <<endl
21.         <<"6: ELFHash 算法" <<endl;
22.     cout<<"请输入整数 1~6 来选择哈希函数 hash_k : " <<endl;
23.     cin>>hash_k;//选择使用的哈希函数
24.     //将所有句子读到 vector<vector<string> >类型的 all 里面,all[k]表示第 k 个句子
25.     bool A1 = _read(all,table,n);//读取 text.txt 并生成对应的 all 和 table
26.     ofstream outfile("answer2.txt",ios::out);//输出到文件 answer.txt 里
27.     cout<<"all sentences: " <<all.size() <<endl;
28.     cout<<"all_words: " <<all_words <<endl;
29.     cout<<"diff_words: " <<diff_words <<endl;
30.     cout<<"search: " <<search <<endl;
31.     cout<<"space = n : " <<n <<endl;
32.     cout<<"n/diff_words: " <<(double)n/(double)diff_words <<endl;
33.     cout<<"average_search = search/diff_words: " << (double)search/(double)diff_words <<endl;
34.     cout<<"-----" <<endl;
35.     cout<<"You can use '$stop' to stop the Query !" <<endl;
36.     cout <<"Query : " <<endl; //测试样例 and 和 amu 在没有解决冲突时 hash 值相同
37.     cout<<"请输入: ";
38.     outfile<<"拉链法: " <<endl;
39.     outfile<<"请输入哈希表的容量 n:(推荐 n>=100) " <<endl <<n <<endl ;
40.     outfile<<"哈希函数选择: " <<endl
41.         <<"1: BKDRHash 算法" <<endl
```

```

42.         <<"2: DJBHash 算法"<<endl
43.         <<"3: JSHash 算法"<<endl
44.         <<"4: RSHash 算法"<<endl
45.         <<"5: SDBMHash 算法"<<endl
46.         <<"6: ELFHash 算法" <<endl;
47.     outfile<<"请输入整数 1~6 来选择哈希函数 hash_k : "<<endl<<hash_k<<endl;
48.     outfile<<"all sentences: "<<all.size()<<endl;
49.     outfile<<"all_words: "<<all_words<<endl;
50.     outfile<<"diff_words: "<<diff_words<<endl;
51.     outfile<<"search: "<<search<<endl;
52.     outfile<<"space = n : "<<n<<endl;
53.     outfile<<"n/diff_words: "<<((double)n)/((double)diff_words<<endl;
54.     outfile<<"average_search = search/diff_words: "<< ((double)search)/((double)diff
    _words <<endl;
55.     outfile<<"-----"<<endl;
56.     outfile<<"You can use '$stop' to stop the Query !"<<endl;
57.     outfile<<"Query :"<<endl;
58.     outfile<<"请输入: ";
59.     string find;
60.     while(cin>>find&&find!="$stop"){//根据输入的单词进行相应操作如查询或退出
61.         outfile<<find<<endl;
62.         int key1 = myhash(find,n);
63.         int key2 = getkey(table,find,n);
64.         if(key2==table[key1].size()){
65.             cout<<"There is not the word!"<<endl;
66.             outfile<<"There is not the word!"<<endl;
67.         }
68.         else{
69.             for(int i = 0 ; i <table[key1][key2]->index.size();++i){
70.                 cout<<"Sentence "<<table[key1][key2]->index[i]+1 <<endl;
71.                 outfile<<"Sentence "<<table[key1][key2]->index[i]+1 <<endl;
72.                 for(int j = 0; j <all[table[key1][key2]->index[i]].size() ;++j){
73.                     cout<<all[table[key1][key2]->index[i]][j]<<" ";
74.                     outfile<<all[table[key1][key2]->index[i]][j]<<" ";
75.                 }
76.                 cout<<endl;
77.                 outfile<<endl;
78.             }
79.         }
80.         cout<<endl<<"-----"<<endl;
81.         cout<<"You can use '$stop' to stop the Query !"<<endl;
82.         cout<< "Query :"<<endl;
83.         cout<<"请输入: ";
84.         outfile<<endl<<"-----"<<endl;
85.         outfile<<"You can use '$stop' to stop the Query !"<<endl;
86.         outfile<< "Query :"<<endl;

```

```

87.     outfile<<"请输入: ";
88. }
89.     outfile.close();
90.     return 0;
91. }

```

3.5 单词平均查找长度计算

定义全局变量：

`int search;` //记录不同单词的查找总次数（总查找长度）

`int diff_words;` //记录不同单词的总数

由于无论是开放地址法还是拉链法，解决冲突时都会调用 `getkey` 函数，则可以在每次有新的单词调用 `getkey` 函数时：计算不同单词总数，不同单词的总查找数。在 `text.txt` 中的数据全部读取时，所有单词都会调用 `getkey` 函数，这样最终的不同单词总数，不同单词的总查找数便计算出了。

则平均查找长度=不同单词的总查找数/不同单词总数。即：

`double average_search = (double)search/(double)diff_words`

```

1.  int getkey(string table[],string s,int n){ //开放地址法
2.  //生成单词 s 在哈希表中已经解决冲突的 key 值
3.      ++all_words;
4.      int key = myhash(s,n); //用哈希函数生成哈希值
5.      int k = 0;
6.      ++search;
7.      while(table[(key+k*k)%n] != "" ){ //平方探测法解决冲突
8.          if(table[(key+k*k)%n] == s ) return (key+k*k)%n;
9.      //如果该单词已经在哈希表 table[] 中时返回该 key 值
10.         ++k;
11.         ++search;
12.     }
13.     ++diff_words;
14.     return (key+k*k)%n; //返回已经解决冲突的 key 值
15. }

```

```

12. int getkey(vector<vector<word*> > &table,string ss,int n){//拉链法
13. //生成单词 s 在哈希表 table[key1]拉链中的索引

```

```

14.    ++all_words;
15.    int key1 = myhash(ss,n);
16.    int h= table[key1].size();
17.    for(int i = 0 ; i < h;++i){ //拉链法
18.        if(table[key1][i]->s == ss) return i;//如果该单词已经在哈希表 table[]中时返回
        该 key 值
19.    }
20.    search+=(h+1);
21.    ++diff_words;
22.    return table[key1].size();
23. }

```

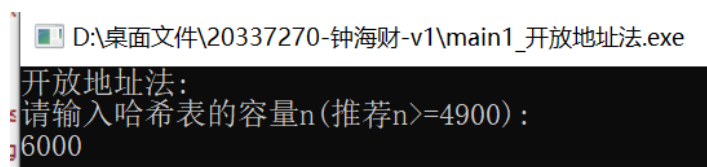
4.程序运行与测试

4.1 开放地址法的运行与测试

4.1.1 开放地址法的程序测试步骤

1.打开 main1_开放地址法.exe

2.程序执行时先输入 n，n 为哈希表的容量，由于是开放地址法，n 推荐大于等于 4900，因为不同的单词总数为 4855：



3.再输入 hash_k，选择使用的哈希函数：

```
D:\桌面文件\20337270-钟海财-v1\main1_开放地址法.exe
开放地址法:
请输入哈希表的容量n(推荐n>=4900):
6000
哈希函数选择:
1: BKDRHash算法
2: DJBHash算法
3: JSHash算法
4: RSHash算法
5: SDBMHash算法
6: ELFHash算法
请输入整数1~6来选择哈希函数hash_k:
3
```

选择完哈希函数就会调用_read 函数，然后输出:

all sentences: 句子总数

all_words: 总单词数

diff_words: 不同单词数

search:不同单词的查找总数

space = n 哈希表的容量

n/diff_word 容量与不同单词数之比

average_search = search/diff_words 平均查找长度

4.然后输入要查询的词 find，可以输入“\$stop”停止查询:

```
D:\桌面文件\20337270-钟海财-v1\main1_开放地址法.exe
开放地址法:
请输入哈希表的容量n(推荐n>=4900):
6000
哈希函数选择:
1: BKDRHash算法
2: DJBHash算法
3: JSHash算法
4: RSHash算法
5: SDBMHash算法
6: ELFHash算法
请输入整数1~6来选择哈希函数hash_k :
3
all sentences: 1728
all_words: 32516
diff_words: 4855
search: 10893
space = n : 6000
n/diff_words: 1.23584
average_search = search/diff_words: 2.24367
-----
You can use '$stop' to stop the Query !
Query :
请输入:
```

5.查看结果：这为查询 American 时的结果，句子序号+句子：

```
You can use '$stop' to stop the Query !
Query :
请输入: American
Sentence 224
About the tenth day of my fame I went to fulfil my duty to my flag by paying my respects to the American minister.
Sentence 243
There was still another guest, an American - but I am a little ahead of my story.
-----
You can use '$stop' to stop the Query !
Query :
请输入:

You can use '$stop' to stop the Query !
Query :
请输入: $stop
-----
Process exited after 344.2 seconds with return value 0
请按任意键继续. . .
```

此为停止时的界面

4.1.2 开放地址法的测试样例及输出

输入：

1. 6000//容量
2. 2//使用的哈希函数
3. stop//以下为查询的词
4. American
5. poor
6. look

```
7. bill
8. spoke
9. gigantic
10. raps
11. above
12. personal
13. first
14. abc
15. return
16. begin
17. $stop//跳出查询
```

输出:

```
D:\桌面文件\20337270-钟海财-v1\main1_开放地址法.exe
开放地址法:
请输入哈希表的容量n(推荐n>=4900):
6000
哈希函数选择:
1: BKDRHash算法
2: DJBHash算法
3: JSHash算法
4: RSHash算法
5: SDBMHash算法
6: ELFHash算法
请输入整数1~6来选择哈希函数hash_k :
2
all sentences: 1728
all words: 32516
diff_words: 4855
search: 11113
space = n : 6000
n/diff_words: 1.23584
average_search = search/diff_words: 2.28898
-----
You can use '$stop' to stop the Query !
Query :
请输入: stop
Sentence 124
Even if I lose their bill, or burn it, they are still safe, because they can stop payment, and the Bank will make them
hole; but meantime I've got to do a month's suffering without wages or profit - unless I help win that bet, whatever it
may be, and get that situation that I am promised.
Sentence 281
You must come and stop with me to-night, when we leave here, and tell me all about it."
Sentence 312
```

 You can use '\$stop' to stop the Query !
 Query :
 请输入: American
 Sentence 224
 About the tenth day of my fame I went to fulfil my duty to my flag by paying my respects to the American minister.
 Sentence 243
 There was still another guest, an American - but I am a little ahead of my story.

 You can use '\$stop' to stop the Query !
 Query :
 请输入:

 You can use '\$stop' to stop the Query !
 Query :
 请输入: poor
 Sentence 32
 They saw many honest faces go by that were not intelligent enough; many that were intelligent, but not honest enough; many that were both, but the possessors were not poor enough, or, if poor enough, were not strangers.
 Sentence 55
 He wanted to look at it, and keep on looking at it; he couldn't seem to get enough of it to quench the thirst of his eyes, but he shrank from touching it as if it had been something too sacred for poor common clay to handle.
 Sentence 101
 We conceive you to be poor and a stranger.
 Sentence 203
 From being a poor, struggling, little hand-to-mouth enterprise, it had become celebrated, and overcrowded with customers.
 Sentence 342
 Henry, it doesn't merely make me realize how rich you are; it makes me realize, to the bone, to the marrow, how poor I am - how poor I am, and how miserable, how defeated, routed, annihilated!"
 Sentence 515
 'Ah, poor dear, so it is!' said the nurse, picking up the cork of the green bottle, which had fallen out on the pillow, as she stooped to take up the child.
 Sentence 534

 You can use '\$stop' to stop the Query !
 Query :
 请输入: look
 Sentence 38
 I was going to open it, but he said no; take it to my lodgings, and look it over carefully, and not be hasty or rash.
 Sentence 55
 He wanted to look at it, and keep on looking at it; he couldn't seem to get enough of it to quench the thirst of his eyes, but he shrank from touching it as if it had been something too sacred for poor common clay to handle.
 Sentence 165
 It's easy to say, sir; but look at the bill yourself."
 Sentence 166
 The proprietor took a look, gave a low, eloquent whistle, then made a dive for the pile of rejected clothing, and began to snatch it this way and that, talking all the time excitedly, and as if to himself: "Sell an eccentric millionaire such an unspeakable suit as that!"
 Sentence 176

 You can use '\$stop' to stop the Query !
 Query :
 请输入: bill
 Sentence 33
 There was always a defect, until I came along; but they agreed that I filled the bill all around; so they elected me unanimously, and there I was now waiting to know why I was called in.
 Sentence 54
 Then he was restored to his normal condition, and made a thousand apologies for not being able to break the bill, and I couldn't get him to touch it.
 Sentence 62
 I was pretty nervous; in fact, pretty badly frightened, though, of course, I was no way in fault; but I knew men well enough to know that when they find they've given a tramp a million-pound bill when they thought it was a one-pounder, they are in a frantic rage against him instead of quarreling with their own near-sightedness, as they ought.
 Sentence 118

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: spoke
Sentence 15
I was just getting desperate enough to brave all the shame, and to seize it, when a window behind me was raised, and
gentleman spoke out of it, saying: "Step in here, please."
Sentence 141
The fellow I spoke to nodded his head towards another fellow, and gave me no answer.
Sentence 480
Of course, I spoke right up now, without any fooling, and went straight to the point.
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: gigantic
Sentence 913
Bumble spoke, he raised his cane to the bill above him, and gave three distinct raps upon the words
were printed thereon in Roman capitals of gigantic size.
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入:
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: raps
Sentence 913
Bumble spoke, he raised his cane to the bill above him, and gave three distinct raps upon the words 'five pounds': which
were printed thereon in Roman capitals of gigantic size.
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入:
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: above
Sentence 211
At first, in these mentions, I was at the bottom of the personal-gossip column; next, I was listed above the knights, ne
xt above the baronets, next above the barons, and so on, and so on, climbing steadily, as my notoriety augmented, until
I reached the highest altitude possible, and there I remained, taking precedence of all dukes not royal, and of all eccl
esiastics except the primate of all England.
Sentence 303
I was so bewitched with her that I couldn't count my hands if they went above a double sequence; and when I struck home
I never discovered it, and started up the outside row again, and would have lost the game every time, only the girl did
the same, she being in just my condition, you see; and consequently neither of us ever got out, or cared to wonder why w
e didn't; we only just knew we were happy, and didn't wish to know anything else, and didn't want to be interrupted.
Sentence 411
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: personal
Sentence 211
At first, in these mentions, I was at the bottom of the personal-gossip column; next, I was listed above the knights, ne
xt above the baronets, next above the barons, and so on, and so on, climbing steadily, as my notoriety augmented, until
I reached the highest altitude possible, and there I remained, taking precedence of all dukes not royal, and of all eccl
esiastics except the primate of all England.
Sentence 1046
It is difficult for a large-headed, small-eyed youth, of lumbering make and heavy countenance, to look dignified under a
ny circumstances; but it is more especially so, when superadded to these personal attractions are a red nose and yellow
smalls.
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: first
Sentence 49
The first thing I noticed, then, was the landlord.
Sentence 130
Pretty soon I was feeling first-rate.
Sentence 199
I took my dinners there, but for breakfast I stuck by Harris's humble feeding house, where I had got my first m
million-pound bill.
Sentence 211
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: abc
There is not the word!
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入:
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: return
There is not the word!
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: begin
Sentence 129
It would begin in a month; after that I should be all right.
Sentence 687
'So you'll begin to pick oakum to-morrow morning at six o'clock,' added the surly one in the white waistcoat.
```

控制台输出结果同步保存到 answer1.txt 里面。

4.2 拉链法的运行与测试

4.2.1 拉链法的程序测试步骤

略，测试步骤与 4.1.1 相同，其中第一步改为 1.打开 main2_拉链法.exe

4.2.2 拉链法的测试样例与输出

输入：

1. 3000//容量
2. 4//使用的哈希函数
3. man//以下为查询的词
4. American
5. poor
6. look
7. spoke
8. traffic
9. office
10. money
11. above
12. personal
13. first

- 14. move
- 15. began
- 16. \$stop//跳出查询

输出:

D:\桌面文件\20337270-钟海财-v1\main2_拉链法.exe

```
拉链法:
请输入哈希表的容量n:(推荐n>=100)
3000
哈希函数选择:
1: BKDRHash算法
2: DJBHash算法
3: JSHash算法
4: RSHash算法
5: SDBMHash算法
6: ELFHash算法
请输入整数1~6来选择哈希函数hash_k :
4
all sentences: 1728
all words: 32516
diff_words: 4855
search: 8795
space = n : 3000
n/diff_words: 0.61792
average_search = search/diff_words: 1.81153
-----
You can use '$stop' to stop the Query !
Query :
请输入: man
Sentence 27
So they went on disputing till Brother B said he would bet twenty thousand pounds that the man would live thirty days, a
nyway, on that million, and keep out of jail, too.
Sentence 31
Then he dictated a letter, which one of his clerks wrote out in a beautiful round hand, and then the two brothers sat at
the window a whole day watching for the right man to give it to.
Sentence 100
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: American
Sentence 224
About the tenth day of my fame I went to fulfil my duty to my flag by paying my respects to the American minister.
Sentence 243
There was still another guest, an American - but I am a little ahead of my story.
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: poor
Sentence 32
They saw many honest faces go by that were not intelligent enough; many that were intelligent, but not honest enough; ma
ny that were both, but the possessors were not poor enough, or, if poor enough, were not strangers.
Sentence 55
He wanted to look at it, and keep on looking at it; he couldn't seem to get enough of it to quench the thirst of his eye
, but he shrank from touching it as if it had been something too sacred for poor common clay to handle.
Sentence 101
We conceive you to be poor and a stranger.
Sentence 203
From being a poor, struggling, little hand-to-mouth enterprise, it had become celebrated, and overcrowded with customers
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: look
Sentence 38
I was going to open it, but he said no; take it to my lodgings, and look it over carefully, and not be hasty or rash.
Sentence 55
He wanted to look at it, and keep on looking at it; he couldn't seem to get enough of it to quench the thirst of his
, but he shrank from touching it as if it had been something too sacred for poor common clay to handle.
Sentence 165
It's easy to say, sir; but look at the bill yourself."
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: spoke
Sentence 15
I was just getting desperate enough to brave all the shame, and to seize it, when a window behind me was raised, and a
gentleman spoke out of it, saying: "Step in here, please."
Sentence 141
The fellow I spoke to nodded his head towards another fellow, and gave me no answer.
Sentence 480
Of course, I spoke right up now, without any fooling, and went straight to the point.
Sentence 504
-----
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: traffic
Sentence 1
When I was twenty-seven years old, I was a mining-broker's clerk in San Francisco, and an expert in all the details of s
tock traffic.
-----
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: office
Sentence 499
The fact is, that there was considerable difficulty in inducing Oliver to take upon himself the office of respiration,--
a troublesome practice, but one which custom has rendered necessary to our easy existence; and for some time he lay gasp
ing on a little flock mattress, rather unequally poised between this world and the next: the balance being decidedly in
favour of the latter.
Sentence 805
When they arrived at the office, he was shut up in a little room by himself, and admonished by Mr.
-----
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: money
Sentence 8
This money fed and sheltered me twenty-four hours.
Sentence 24
Well, the brothers, chatting along, happened to get to wondering what might be the fate of a perfectly honest and intell
igent stranger who should be turned adrift in London without a friend, and with no money but that million-pound bank-not
e, and no way to account for his being in possession of it.
Sentence 41
As soon as I was out of sight of that house I opened my envelope, and saw that it contained money!
-----
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: above
Sentence 211
At first, in these mentions, I was at the bottom of the personal-gossip column; next, I was listed above the knights, n
xt above the baronets, next above the barons, and so on, and so on, climbing steadily, as my notoriety augmented, until
I reached the highest altitude possible, and there I remained, taking precedence of all dukes not royal, and of all ecc
lesiastics except the primate of all England.
Sentence 303
I was so bewitched with her that I couldn't count my hands if they went above a double sequence; and when I struck home
I never discovered it, and started up the outside row again, and would have lost the game every time, only the girl did
the same, she being in just my condition, you see; and consequently neither of us ever got out, or cared to wonder why
e didn't; we only just knew we were happy, and didn't wish to know anything else, and didn't want to be interrupted.
-----
```

```
-----
You can use '$stop' to stop the Query !
Query :
请输入: personal
Sentence 211
At first, in these mentions, I was at the bottom of the personal-gossip column; next, I was listed above the knights, n
xt above the baronets, next above the barons, and so on, and so on, climbing steadily, as my notoriety augmented, until
I reached the highest altitude possible, and there I remained, taking precedence of all dukes not royal, and of all ecc
lesiastics except the primate of all England.
Sentence 1046
It is difficult for a large-headed, small-eyed youth, of lumbering make and heavy countenance, to look dignified under a
ny circumstances; but it is more especially so, when superadded to these personal attractions are a red nose and yellow
smalls.
Sentence 1211
Sowerberry and Bumble, being personal friends of the clerk, sat by the fire with him, and read the paper.
Sentence 1248
-----
```

```

You can use '$stop' to stop the Query !
Query :
请输入: first
Sentence 49
The first thing I noticed, then, was the landlord.
Sentence 130
Pretty soon I was feeling first-rate.
Sentence 199
I took my dinners there, but for breakfast I stuck by Harris's humble feeding house, where I had got my first meal on my
million-pound bill.
Sentence 211

```

```

You can use '$stop' to stop the Query !
Query :
请输入: move
Sentence 13
But every time I made a move to get it some passing eye detected my purpose, and of course I straightened up then, and l
ooked indifferent, and pretended that I hadn't been thinking about the pear at all.

```

```

You can use '$stop' to stop the Query !
Query :
请输入: began
Sentence 34
They began to ask me questions about myself, and pretty soon they had my story.
Sentence 63
As I approached the house my excitement began to abate, for all was quiet there, which made me feel pretty sure the blun
der was not discovered yet.
Sentence 127
My hopes began to rise high.
Sentence 166

```

控制台输出结果同步保存到 answer2.txt 里面。

5.实验总结与心得

（一）开放地址法的平均查找长度与哈希表容量，哈希函数的关系

哈希表容量与不同单词总数之比 (n/diff_words)	1.02987	1.23584	1.85376	3.0896	6.1792	10.2987
BKDRHash 算法	4.16849	2.28465	1.52976	1.23337	1.09866	1.05891
DJBHash 算法	4.34233	2.28898	1.51452	1.22183	1.09846	1.05149
JSHash 算法	4.21668	2.24367	1.49866	1.22678	1.09022	1.05561
RSHash 算法	3.85644	2.26962	1.49289	1.21833	1.09949	1.05953
SDBMHash 算法	4.03872	2.22657	1.52606	1.22987	1.0964	1.0585
ELFHash 算法	太慢	太慢	1.70525	1.3139	1.22801	1.12049

由得到的上表中的数据可知，使用开发地址法解决冲突时，除 ELEHash 外

的几种哈希函数的平均查找度都差不多，哈希值分布都比较均匀，但 ELEHash 算法的平均查找度显然更大，可能是算法的原因，也可能是测试文本的偶然性的原因。

且对于任一种哈希函数，哈希表容量与不同单词总数之比越大时，平均查找长度越小，且趋近于 1。但容量接近于不同单词总数时，平均查找长度显著增大，有的算法甚至运行时间超级长（ELFHash 算法）。

（二）拉链法的平均查找长度与哈希表容量，哈希函数的关系

哈希表容量与不同单词总数之比 (n/diff_words)	0.411946	1.02987	1.85376	3.0896	6.1792	10.2987
BKDRHash 算法	2.20783	1.47024	1.28198	1.16251	1.08321	1.05211
DJBHash 算法	2.22513	1.48177	1.27683	1.15366	1.08157	1.04593
JSHash 算法	2.20247	1.47971	1.27106	1.15778	1.07497	1.04882
RSHash 算法	2.22801	1.48877	1.27477	1.15922	1.08404	1.0517
SDBMHash 算法	2.24531	1.4828	1.27765	1.16231	1.08136	1.05211
ELFHash 算法	3.73471	1.67827	1.37817	1.22533	1.18414	1.10257

从上表中数据发现，使用拉链法解决冲突和使用开放地址法解决冲突一样：除 ELEHash 外的几种哈希函数的平均查找度都差不多，哈希值分布都比较均匀，但 ELEHash 算法的平均查找度显然更大，这里我推断应该是 ELEHash 算法生成的哈希值分布没有其它哈希函数均匀。

同时，将拉链法和开放地址法间进行对比，显然在相同容量时，拉链法的平均查找长度比开放地址法的平均查找长度更短。且在容量越小越接近不同单

词总数时，拉链法的平均查找长度显著小于开放地址法；在容量越大时，拉链法的平均查找长度虽仍小于开放地址法，但这时两者之间的差距越来越小。

（三）总结

（1）本次实验测试的 6 种哈希函数：

1: BKDRHash 算法, 2: DJBHash 算法, 3: JSHash 算法,
4: RSHash 算法, 5: SDBMHash 算法, 6: ELFHash 算法。

前 5 种哈希函数的平均查找长度都很接近，哈希值分布较均匀；只有第 6 种哈希函数 ELFHash 的平均查找长度显然更大，哈希值分布较不均匀。

（2）拉链法的平均查找长度小于开放地址法，且容量越小时越显著。所以容量较小时推荐使用拉链法，容量较大时两者差距不大，可根据需要选择。

附录、提交文件清单

实验报告 1 份：	20337270-钟海财-v1.doc
测试文本：	text.txt
哈希函数文件：	myhash.cpp
程序文件：	
	main1_开放地址法.cpp
	main1_开放地址法.exe
	main2_拉链法.cpp
	main2_拉链法.exe
输出存放文件：	answer1.txt, answer2.txt