

实验题目：

有序数组，（单）链表，红黑树的插入、查找、删除效率比较。

实验背景：

学校有100000台主机，每台主机有一个mac地址

学校将100000台主机编号（index）为（1-100000）

现考虑使用有序数组、（单）链表和红黑树三种数据结构来存储并管理主机编号（index）及其mac地址

实验内容：

1. 按照代码框架，自行实现数组、链表和红黑树的插入、查找、删除算法。
2. （插入）对于每种数据结构，首先依次插入99900个数据项，然后依次插入100个数据项，对后者计时。
3. （查找）对于每种数据结构，（由2知100000个表项已经完全插入），给出不少于十个index（尽量均匀），查询相应的mac地址值并打印。对查找过程计时。
4. （删除）对于每种数据结构，**根据index**（尽量均匀）依次删除不少于10个数据项，并输出相应的mac地址值。对删除过程计时。

实验细节：

1. **语言不限**，但助教会提供C++版本的一个简单框架，具体函数需要同学们自己实现。使用C++的同学务必在框架内实现具体功能。使用其他语言的同学也要尽量保证函数接口一致和保证输出格式（见附录）一致。（上课学到的是老师PPT上的算法，硬要按自己的算法实现的话相当于这门课没有收获）
2. 助教提供data.csv文件，里面包含100000个主机的index和mac（mac地址均以9A开头），助教提供一种IO方式，同学们亦可自行实现。
3. 检查时助教会随机挑选若干index，以验证查找算法的正确性。（比如助教让你查找index为10的主机的mac地址，需输出"9ADF46C59C78"
4. 实验难点在于红黑树，遇到困难的同学应多参照PPT和课本的标准算法，自行实现。
5. 编译出的文件大多数情况下应为命令行可执行文件，windows下是exe，可使用cmd或powershell运行，macos和Linux下是可执行文件，可使用bash等shell运行（注意Unix环境下计时windows.h不可用，计时方法等需要自己实现）。
6. 源文件和头文件的编码是GB2312，主函数文件的编码是UTF-8 with BOM。中文乱码的同学请注意。

附录A：有序数组，链表及平衡二叉树的插入、查找、删除的时间复杂度

	有序数组	链表	(平衡) 二叉查找树
搜索	$O(\log(n))$	$O(n)$	$O(\log(n))$
删除	$O(n)$	$O(n)$	$O(\log(n))$
插入	$O(n)$	$O(1)$	$O(\log(n))$

附录B：示例框架在VStudio中测试通过，下图为部分输出样例。

```
There are 100000 datas in the CSV file.
ARRAY_insert_time = 0.0710549 s
10000 : 9AB9AB17AE20
20000 : 9AD882878B42
30000 : 9AFD2A21BA0A
40000 : 9AEAEA63FA4E
50000 : 9A3F7E3D3D71
60000 : 9ABE5D536874
70000 : 9AE37EAAAC282
80000 : 9AA14CCCA53B
90000 : 9A49EA495138
99999 : 9AFAEC8EE6F5
ARRAY_query_time = 0.0082415 s
Start ARRAY_Delete: 100000
15000 : 9A56D43B52EB
25000 : 9A3DEE4DECE5
35000 : 9A4606D7D749
45000 : 9AAEF2A75E11
55000 : 9A262ED63A7D
65000 : 9A7222B53C39
75000 : 9A3108ECDD64
85000 : 9A5E12F386CF
95000 : 9A53CA07DA7B
99999 : 9AFAEC8EE6F5
ARRAY_delete_time = 0.013876 s

LINKED_insert_time = 1.1e-05 s
10000 : 9AB9AB17AE20
20000 : 9AD882878B42
30000 : 9AFD2A21BA0A
40000 : 9AEAEA63FA4E
50000 : 9A3F7E3D3D71
60000 : 9ABE5D536874
70000 : 9AE37EAAAC282
80000 : 9AA14CCCA53B
90000 : 9A49EA495138
99999 : 9AFAEC8EE6F5
LINKED_query_time = 0.0189491 s
Start LINKED_Delete:
15000 : 9A56D43B52EB
25000 : 9A3DEE4DECE5
35000 : 9A4606D7D749
45000 : 9AAEF2A75E11
55000 : 9A262ED63A7D
65000 : 9A7222B53C39
75000 : 9A3108ECDD64
85000 : 9A5E12F386CF
95000 : 9A53CA07DA7B
99999 : 9AFAEC8EE6F5
LINKED_Delete_time = 0.0202646 s

RBTREE_insert_time = 0.0001198 s
10000 : 9AB9AB17AE20
20000 : 9AD882878B42
30000 : 9AFD2A21BA0A
40000 : 9AEAEA63FA4E
50000 : 9A3F7E3D3D71
60000 : 9ABE5D536874
70000 : 9AE37EAAAC282
80000 : 9AA14CCCA53B
90000 : 9A49EA495138
99999 : 9AFAEC8EE6F5
RBTREE_Query_time = 0.0085943 s
RBTREE_Delete time = 3.67e-05 s
```

```

RBTree test
8(B) is root
8(B) is root
2(R) is 8's left child
5(B) is root
2(R) is 5's left child
8(R) is 5's right child
5(B) is root
2(B) is 5's left child
8(B) is 5's right child
9(R) is 8's right child
5(B) is root
2(B) is 5's left child
1(R) is 2's left child
8(B) is 5's right child
9(R) is 8's right child
5(B) is root
2(B) is 5's left child
1(R) is 2's left child
8(B) is 5's right child
7(R) is 8's left child
9(R) is 8's right child
5(B) is root
2(B) is 5's left child
1(R) is 2's left child
8(R) is 5's right child
7(B) is 8's left child
6(R) is 7's left child
9(B) is 8's right child
5(B) is root
2(B) is 5's left child
1(R) is 2's left child
8(R) is 5's right child
7(B) is 8's left child
6(R) is 7's left child
9(B) is 8's right child
10(R) is 9's right child
5(B) is root
2(B) is 5's left child
1(R) is 2's left child
4(R) is 2's right child
8(R) is 5's right child
7(B) is 8's left child
6(R) is 7's left child
9(B) is 8's right child
10(R) is 9's right child
5(B) is root
4(B) is 5's left child
1(R) is 4's left child
8(R) is 5's right child
7(B) is 8's left child
6(R) is 7's left child
9(B) is 8's right child
10(R) is 9's right child

```

#### 附录C：运行命令行应用程序的方法

1. 在代码中配置好csv文件的路径，将csv文件放置到对应路径。
2. 使用命令行运行应用程序

```
(base) PS D:\Doc\算法导论\lab2\lab2release\Lab2_51\x64\Release> ls
```

```
Directory: D:\Doc\算法导论\lab2\lab2release\Lab2_51\x64\Release
```

Mode	LastWriteTime		Length	Name
----	-----		-----	----
d----	2021/5/5	14:41		Lab2.tlog
-a---	2021/5/5	14:41	1010062	array.obj
-a---	2021/4/27	15:11	1988904	data_100000.csv
-a---	2021/5/5	14:41	45568	Lab2.exe
-a---	2021/5/5	14:41	308	Lab2.exe.recipe
-a---	2021/5/5	14:41	586917	Lab2.iobj
-a---	2021/5/5	14:41	325216	Lab2.ipdb
-a---	2021/5/5	14:41	364	Lab2.log
-a---	2021/5/5	14:41	1365150	Lab2.obj
-a---	2021/5/5	14:41	1011712	Lab2.pdb
-a---	2021/5/5	14:41	67	Lab2.vcxproj.FileListAbsolute.txt
-a---	2021/5/5	14:41	1006674	linked.obj
-a---	2021/5/5	14:41	1267858	RBTree.obj
-a---	2021/5/5	14:41	1118059	ultity.obj
-a---	2021/5/5	14:41	643072	vc142.pdb
-a---	2021/5/5	14:41	2	vcpkg.applocal.log

```
(base) PS D:\Doc\算法导论\lab2\lab2release\Lab2_51\x64\Release> ./lab2.exe
```

```
There are 100000 datas in the CSV file.
```