

算法基础 实验三 红黑树和区间树

PB18000334 徐家恒

一、实验内容

红黑树

实现红黑树的基本算法，分别对整数 $n=20、40、60、80、100$ ，随机生成 n 个互异的正整数 ($K_1, K_2, K_3, \dots, K_n$)，以这 n 个正整数作为结点的关键字，向一棵初始空的红黑树中依次插入 n 个节点，统计算法运行所需时间，画出时间曲线。

随机删除红黑树中 $n/4$ 个结点，统计删除操作所需时间，画出时间曲线图。

区间树

实现区间树的基本算法，随机生成 30 个正整数区间，以这 30 个正整数区间的左端点作为关键字构建红黑树，向一棵初始空的红黑树中依次插入 30 个节点，然后随机选择其中 3 个区间进行删除。实现区间树的插入、删除、遍历和查找算法。

二、实验设备

PC

三、实验环境

Visual Studio 2019

四、实验方法

红黑树:

```
struct RBT {
    RBT *left = nullptr, *right = nullptr, *p = nullptr;
    int key = 0;
    int color = BLACK;
};

RBT *nil = new RBT{ nullptr, nullptr, nullptr, 0, BLACK };
typedef RBT *node;
```

用到的函数:

```
//旋转树
void leftRotate(node&, node);
void rightRotate(node&, node);
//插入
void RBInsert(node&, node);
void RBInsertFixup(node&, node);
//删除
void RBTranslant(node&, node, node);
node TreeMinimum(node);
void RBDelete(node&, node);
void RBDeleteFixup(node&, node);
//初始化
inline int rand(int);
node initRBT(int, std::istream&, std::ostream&, std::ostream&, std::ostream&, std::ostream&);
//遍历
void inOrder(node, std::ostream&);
Data traversal(node, int, int);
void gotoxy(short, short, int digit = 1);
```

区间树:

```
struct Interval {
    int low = -1, high = -1;
};

struct SegRBT {
    SegRBT *left = nullptr, *right = nullptr, *p = nullptr;
    Interval interval;
    int imax = 0;
    int color = BLACK;
};

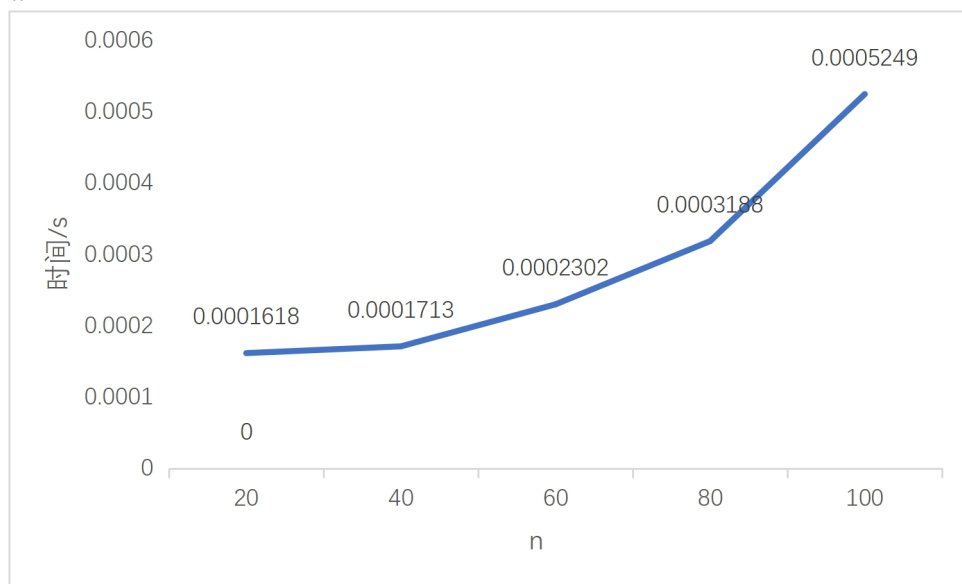
SegRBT *nil = new SegRBT{ nullptr, nullptr, nullptr, -1, -1, -1, BLACK };
typedef SegRBT *node;
typedef int Color;
```

函数基本同红黑树

五、结果分析

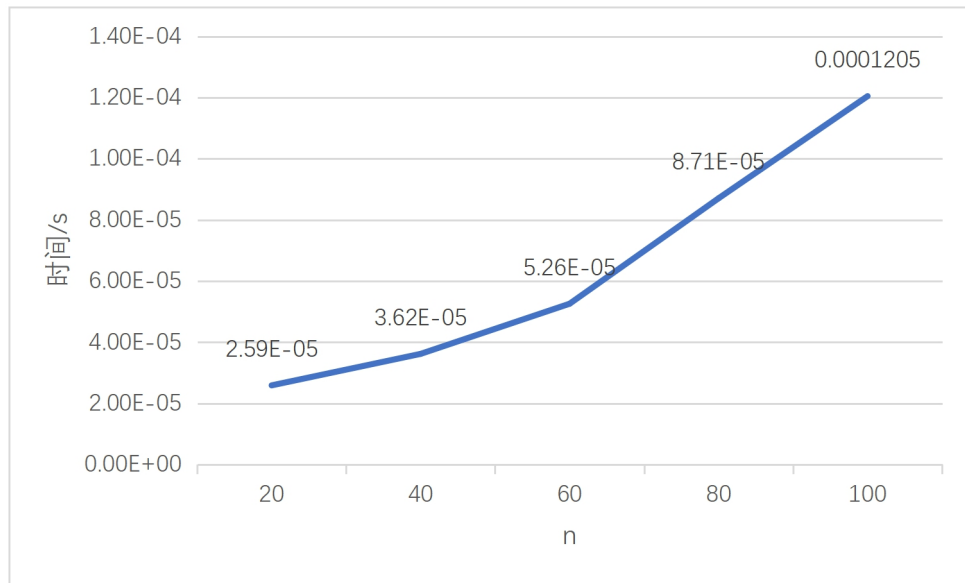
红黑树:

插入



符合 $n \lg n$

删除



符合 $n \lg n$

区间树:

搜索结果: (-1, -1) 区间为 nil

```
search.txt - Notepad
File Edit Format View Help
2 24
-1 -1
30 39
```