实验6: 查找

确保把类定义写在`mySearch.h`中，类实现写在`mySearch.cpp`中，**不要修改文件名！！！**

`mySearch.h`文件中已经给出类定义和必要的函数，可以自行添加需要用到的函数，但不要变更已有的函数声明，测试文件只会调用已有的函数。

最终提交两个文件：`mySearch.h`和`mySearch.cpp`

注意：请务必使用在linux g++ std=c++11编译环境下可以使用的库函数

- 实现顺序查找表和它的二分查找方法。顺序查找表类`SSTable`，采用线性表存储，完成以下功能：
  - 顺序查找表的建立`SSTable(int, int*)`，参数列表为：
    - 元素数目`int`
    - 元素列表`int*`，输入数据已按升序排列，不包含重复值
  - 顺序查找表的二分查找方法`int binSearch(int)`，参数为要查找的元素值，若找到元素，返回对应元素在顺序查找表中的下标（**从0开始**）；若未找到元素，返回`-1`
  - 其它类初始化、销毁、私有属性存取等基本功能
- 实现二叉排序树和它的查找、新增节点、删除节点方法。二叉排序树节点类`BSTreeNode`、二叉排序树类`BSTree`，完成以下功能：
  - 二叉排序树的建立`BSTree(int, int*)`，参数列表为：
    - 元素数目`int`
    - 元素列表`int*`，输入数据不包含重复值
  - 二叉排序树的遍历输出`string printTree()`，返回二叉排序树的**前序**序列string类型字符串，格式为：

    ```
    “顶点值+空格”（序列最后一个顶点后也要添加空格）

    例如：“3 4 5 6 ”（示例仅为格式参考，不是一棵树）
    ```

  - 二叉排序树的查找`bool searchNode(int)`，参数为要查找的元素值。若找到返回`true`，未找到返回`false`
  - 二叉排序树添加节点`bool addNode(int)`，参数为要添加的元素值。若添加成功返回`true`，若该元素与已有元素值重复，返回`false`
  - 二叉排序树删除节点`bool deleteNode(int)`，参数为要删除的元素值。若删除成功返回`true`，若该元素不存在，返回`false`
  - 其它类初始化、销毁、私有属性存取等基本功能
  - 提示：可以使用递归方法，自行添加需要用到的辅助函数，根据自己的习惯自定义实现算法的中间函数，但请注意不要出现编译错误。

输入输出示例：

```cpp
#include <iostream>
#include "myGraph.h"

using namespace std;

```

```cpp
int main()
{
    int n1 = 6;
    int data1[6] = {1, 2, 4, 5, 6, 9};
    SSTable s(n1, data1);
    cout << s.binSearch(4) << endl;
    cout << s.binSearch(7) << endl;

    int n2 = 5;
    int data2[5] = {3, 1, 2, 5, 4};
    BSTree b(n2, data2);

    cout << b.printTree() << endl;
    cout << b.searchNode(2) << endl;
    cout << b.searchNode(7) << endl;

    cout << b.addNode(7) << endl;
    cout << b.addNode(7) << endl;
    cout << b.deleteNode(2) << endl;
    cout << b.deleteNode(2) << endl;

    cout << b.printTree() << endl;
    cout << b.searchNode(2) << endl;
    cout << b.searchNode(7) << endl;

    return 0;
}

/*
    控制台运行结果为：
    2
    -1
    3 1 2 5 4
    1
    0
    1
    0
    1
    0
    3 1 5 4 7
    0
    1
*/
```