```
/* BST.h
 *
 * Binary Search Tree class Interface.
 */

//file guards
#ifndef _BST_H_
#define _BST_H_

#include "listT.cpp"

class BinarySearchTree
{
 private:
  typedef struct BSTreeNode
  {
    string showName; // title of TV show
    int startDate; // year of TV show release
    int endDate; // year that TV show ended
    string genre; // genre of Tv show
    string url; // url that links to find TV show info
    LinkedList L; // list object that stores actor names for each TV show

    BSTreeNode *leftPtr; // pointer to left subtree
    BSTreeNode *rightPtr; // pointer to right subtree
  } *TreePtr;
  TreePtr rootPtr; // root of the BST

  // if no BST, declare the root NULL
  void InitBSTree()
  { rootPtr = NULL; }

  // search BST by TV show name, print out actors per show
  void SearchNodeInBST( TreePtr treePtr,
               string newShowName );

  // search BST by actor name, prints out each actor in
  void ActorSearchNodeInBST( TreePtr treePtr,
                    string newActorName );

  // search BST, prints shows by decade TV show released in
  void DecadeSearchNodeInBST( TreePtr treePtr,
                     int releaseDate,
                     int stopDate );

  // print TV show names in alphabetical (but only by first letter) order
  void PrintBST_InOrder( TreePtr treePtr );

 public:
  BinarySearchTree() { InitBSTree(); }

  // reads in an external TV show file
  void readFile();

  // adds in new node to BST
  void AddNode( int newStartDate2,
           string showName2,
           int endDate2,
           string genre2,
           string url2,
           LinkedList L12 );

  // public versions of above functs
  void SearchNode( string newShowName );
  void ActorSearchNode( string newActorName );
  void DecadeSearchNode( int releaseDate,
                   int stopDate );

  void PrintInOrder();
};
#endif
```