```
1
2  // COS30008, Problem Set 4, Problem 3, 2022
3
4  #pragma once
5
6  #include "BinarySearchTree.h"
7
8  #include <stack>
9
10 template<typename T>
11 class BinarySearchTreeIterator
12 {
13 private:
14
15     using BSTree = BinarySearchTree<T>;
16     using BNode = BinaryTreeNode<T>;
17     using BTreeNode = BNode*;
18     using BTNStack = std::stack<BTreeNode>;
19
20     const BSTree& fBSTree;      // binary search tree
21     BTNStack fStack;            // DFS traversal stack
22
23     void pushLeft(BTreeNode aNode) {
24         while (!aNode->empty())
25         {
26             if (!fBSTree.fRoot->empty())
27             {
28                 fStack.push(aNode); // pushes aNode to the root
29                 fBSTree.fRoot = fBSTree.fRoot->left;  // makes root = left node
30             }
31         }
32
33     }
34
35 public:
36
37     using Iterator = BinarySearchTreeIterator<T>;
38
39     BinarySearchTreeIterator(const BSTree& aBSTree)
40         :
41         fBSTree(&BNode::NIL),
42         fStack(&BNode::NIL)
43     {
44
45     }
46
47     const T& operator*() const //get key
48     {
49         if (fStack.empty() == false)
```

```
50          {
51              return fStack.top();
52          }
53      }
54
55      Iterator& operator++()
56      {
57          if ( fStack.empty() == false)
58          {
59              fStack.pop();
60              //fBSTree = fBSTree.fRoot->right;
61              return *this;
62          }
63      }
64
65      Iterator operator++(int)
66      {
67          BinarySearchTreeIterator temp = *this;
68          ++(*this);
69          return temp;
70      }
71
72      bool operator==( const Iterator& aOtherIter ) const
73      {
74          return fStack.size() == aOtherIter.fStack.size();
75      }
76
77      bool operator!=(const Iterator& aOtherIter) const
78      {
79          return !(*this == aOtherIter);
80      }
81
82      Iterator begin() const {
83          return BinarySearchTreeIterator<T>(*this);
84      }
85      Iterator end() const
86      {
87          return begin().end();
88      }
89  };
90
```