BST<int,string> c;  
 c.insert(10,"a");  
 c.insert(9,"z");  
 c.insert(11,"b");  
 cout << c.search(10)->value.first << ", " << c.search(10)->value.second << endl;  
 cout << c.search(11)->value.first << ", " << c.search(11)->value.second << endl;  
 cout << c.search(9)->value.first << ", " << c.search(9)->value.second << endl;  
 c.BSTdelete(10);  
 cout << c.search(10)->value.first << ", " << c.search(10)->value.second << endl;  
 c.insert(10,"a");  
 cout << c.search(10)->value.first << ", " << c.search(10)->value.second << endl;  
  
  
  
 TFT<int,string> d;  
 d.insert(10,"a");  
 cout << d.search(10).first << ", " << d.search(10).second << endl;  
 d.insert(11,"b");  
 cout << d.search(11).first << ", " << d.search(11).second << endl;  
 d.insert(9,"c");  
 cout << d.search(9).first << ", " << d.search(9).second << endl;  
 d.insert(12,"d");  
 cout << d.search(12).first << ", " << d.search(12).second << endl;  
 d.TFTdelete(12);  
 cout << d.search(12).first << ", " << d.search(12).second << endl;  
  
 BST<int,int>\* a = new BST<int,int>;  
 TFT<int,int>\* b = new TFT<int,int>;  
 auto start\_time = chrono::high\_resolution\_clock::now();  
 cout << "Tests for Binary Search Tree" << endl;  
  
 cout << "Increasing" << endl;  
  
 for(int i = 1; i <= 10000;i++)  
 {  
 start\_time = chrono::high\_resolution\_clock::now();  
 a->insert(i,i);  
 if(i == 10)  
 cout << "Time for 10th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==100)  
 cout << "Time for 100th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==1000)  
 cout << "Time for 1000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 }  
 cout << "Time for 10000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
  
  
 cout <<"Delete Increasing" << endl;  
 for(int i = 1; i <= 10000;i++)  
 {  
 start\_time = chrono::high\_resolution\_clock::now();  
 a->BSTdelete(i);  
 if(i == 10)  
 cout << "Time for 10th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==100)  
 cout << "Time for 100th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==1000)  
 cout << "Time for 1000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
  
 }  
 cout << "Time for 10000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 delete a;  
 a = new BST<int,int>;  
 cout << "Decreasing" << endl;  
 for(int i = 0; i < 10000;i++)  
 {  
 start\_time = chrono::high\_resolution\_clock::now();  
 a->insert(10000-i,1);  
 if(i == 10)  
 cout << "Time for 10th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==100)  
 cout << "Time for 100th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==1000)  
 cout << "Time for 1000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
  
 }  
 cout << "Time for 10000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
  
 cout <<"Delete Decreasing" << endl;  
 for(int i = 0; i < 10000;i++)  
 {  
 start\_time = chrono::high\_resolution\_clock::now();  
 a->BSTdelete(10000-i);  
 if(i == 10)  
 cout << "Time for 10th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==100)  
 cout << "Time for 100th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==1000)  
 cout << "Time for 1000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
  
 }  
 cout << "Time for 10000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
  
 delete a;  
 a = new BST<int,int>;  
 int hold[10000];  
 cout << "Random" << endl;  
 for(int i = 0; i < 10000;i++)  
 {  
 start\_time = chrono::high\_resolution\_clock::now();  
 hold[i] = rand()%10000+1;  
 a->insert(hold[i],1);  
 if(i == 10)  
 cout << "Time for 10th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==100)  
 cout << "Time for 100th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==1000)  
 cout << "Time for 1000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
  
 }  
 cout << "Time for 10000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
/\*  
 cout << "Delete Random" << endl;  
 for(int i = 1; i <= 10000;i++)  
 {  
 start\_time = chrono::high\_resolution\_clock::now();  
 a->BSTdelete(hold[i]);  
 if(i == 10)  
 cout << "Time for 10th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==100)  
 cout << "Time for 100th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==1000)  
 cout << "Time for 1000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
  
 }  
 cout << "Time for 10000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
\*/  
 cout << "Tests for 2-4 Tree" << endl;  
 delete a;  
  
 cout << "Increasing" << endl;  
 for(int i = 1; i <= 100000;i++)  
 {  
 start\_time = chrono::high\_resolution\_clock::now();  
 b->insert(i,i);  
 if(i == 10)  
 cout << "Time for 10th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==100)  
 cout << "Time for 100th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==1000)  
 cout << "Time for 1000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==10000)  
 cout << "Time for 10000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
  
 }  
 cout << "Time for 100000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
/\*  
 cout << "Delete Increasing" << endl;  
 for(int i = 1; i <= 100000;i++)  
 {  
 start\_time = chrono::high\_resolution\_clock::now();  
 b->TFTdelete(i);  
 if(i == 10)  
 cout << "Time for 10th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==100)  
 cout << "Time for 100th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==1000)  
 cout << "Time for 1000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==10000)  
 cout << "Time for 10000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
  
 }  
 cout << "Time for 100000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
\*/  
 delete b;  
 b = new TFT<int,int>;  
 cout << "Decreasing" << endl;  
 for(int i = 0; i < 100000;i++)  
 {  
 start\_time = chrono::high\_resolution\_clock::now();  
 b->insert(100000-i,i);  
 if(i == 10)  
 cout << "Time for 10th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==100)  
 cout << "Time for 100th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==1000)  
 cout << "Time for 1000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==10000)  
 cout << "Time for 10000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
  
 }  
 cout << "Time for 100000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
/\*  
 cout << "Delete Decreasing" << endl;  
 for(int i = 0; i < 100000;i++)  
 {  
 start\_time = chrono::high\_resolution\_clock::now();  
 b->TFTdelete(100000-i);  
 if(i == 10)  
 cout << "Time for 10th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==100)  
 cout << "Time for 100th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==1000)  
 cout << "Time for 1000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==10000)  
 cout << "Time for 10000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
  
 }  
 cout << "Time for 100000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
\*/  
 cout << "Random" << endl;  
 for(int i = 0; i < 10000;i++)  
 {  
 start\_time = chrono::high\_resolution\_clock::now();  
 hold[i] = rand()%10000+1;  
 b->insert(hold[i],i);  
 if(i == 10)  
 cout << "Time for 10th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==100)  
 cout << "Time for 100th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==1000)  
 cout << "Time for 1000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 }  
 cout << "Time for 10000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
  
 cout << "Delete Random" << endl;  
 for(int i = 0; i < 10000;i++)  
 {  
 start\_time = chrono::high\_resolution\_clock::now();  
 b->TFTdelete(hold[i]);  
 if(i == 10)  
 cout << "Time for 10th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==100)  
 cout << "Time for 100th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 else if(i==1000)  
 cout << "Time for 1000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
 }  
 cout << "Time for 10000th: " << chrono::duration\_cast<chrono::nanoseconds>(chrono::high\_resolution\_clock::now()-start\_time).count() << endl;  
}

Sample output:

10, a

11, b

9, z

0,

10, a

10, a

11, b

9, c

12, d

0,

Tests for Binary Search Tree

Increasing

Time for 10th: 1100

Time for 100th: 2300

Time for 1000th: 17900

Time for 10000th: 158500

Delete Increasing

Time for 10th: 800

Time for 100th: 2000

Time for 1000th: 15300

Time for 10000th: 166800

Decreasing

Time for 10th: 1300

Time for 100th: 2200

Time for 1000th: 16300

Time for 10000th: 152000

Delete Decreasing

Time for 10th: 700

Time for 100th: 1700

Time for 1000th: 14700

Time for 10000th: 162100

Random

Time for 10th: 1200

Time for 100th: 1400

Time for 1000th: 1600

Time for 10000th: 9100

Tests for 2-4 Tree

Increasing

Time for 10th: 1100

Time for 100th: 1700

Time for 1000th: 1600

Time for 10000th: 2600

Time for 100000th: 5200

Decreasing

Time for 10th: 700

Time for 100th: 600

Time for 1000th: 500

Time for 10000th: 1300

Time for 100000th: 4600

Random

Time for 10th: 700

Time for 100th: 600

Time for 1000th: 600

Time for 10000th: 2000

Delete Random

Time for 10th: 800

Time for 100th: 600

Time for 1000th: 600

Time for 10000th: 4000

The beginning section was used to test out each of the functions of both the BST and (2,4) tree. On a small scale all of the functions worked, but on the larger scale tested in the experiments afterwards, the delete function did not work for random inputs in a BST and the delete function for the (2,4) tree would not work for very large increasing or decreasing samples.