```
Node* insert_aux(Node* root, int value) {
Node* o;
  if (root == null) {
    o = new Node(value);
  else {
    if (root.value == value) {
      o = root;
    else {
      if (value < root.value) {</pre>
        root.left = insert_aux(root.left, value);
        root.left = balance(root, LEFT);
        o = root;
                                                                 root
      else {
        // Symmetrical
  return o;
```

0

```
Node* insert_aux(Node* root, int value) {
  Node* o;
  if (root == null) {
  o = new Node(value);
  else {
    if (root.value == value) {
      o = root;
    else {
      if (value < root.value) {</pre>
        root.left = insert_aux(root.left, value);
        root.left = balance(root, LEFT);
        o = root;
                                                               root = null
      else {
        // Symmetrical
                                                                    BT(h=0)
  return o;
```

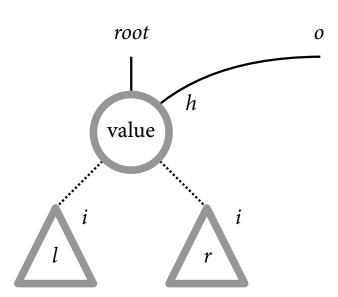
```
Node* insert_aux(Node* root, int value) {
  Node* o;
  if (root == null) {
  →o = new Node(value);
  else {
    if (root.value == value) {
      o = root;
    else {
      if (value < root.value) {</pre>
        root.left = insert_aux(root.left, value);
        root.left = balance(root, LEFT);
        o = root;
                                                                root = null
      else {
                                                                                          RT(0)
        // Symmetrical
                                                            BT(h=0)
                                                                                     value
                                                                                            BT(0)
                                                                             BT(0)
  return o;
```

```
Node* insert_aux(Node* root, int value) {
  Node* o;
  if (root == null) {
    o = new Node(value);
  else {
  if (root.value == value) {
      o = root;
    else {
      if (value < root.value) {</pre>
        root.left = insert_aux(root.left, value);
        root.left = balance(root, LEFT);
        o = root;
                                                                  root
      else {
        // Symmetrical
                                                                        h
  return o;
```

0

```
Node* insert_aux(Node* root, int value) {
  Node* o;
  if (root == null) {
    o = new Node(value);
  else {
    if (root.value == value) {
     o = root;
    else {
      if (value < root.value) {</pre>
        root.left = insert_aux(root.left, value);
        root.left = balance(root, LEFT);
                                                                  root
                                                                                     0
      else {
        // Symmetrical
                                                                         h
                                                                  value
  return o;
```

```
Node* insert_aux(Node* root, int value) {
  Node* o;
  if (root == null) {
    o = new Node(value);
  else {
    if (root.value == value) {
    o = root;
    else {
      if (value < root.value) {</pre>
        root.left = insert_aux(root.left, value);
        root.left = balance(root, LEFT);
        o = root;
      else {
        // Symmetrical
  return o;
```



```
Node* insert_aux(Node* root, int value) {
  Node* o;
  if (root == null) {
    o = new Node(value);
  else {
    if (root.value == value) {
      o = root;
    else {
    if (value < root.value) {
        root.left = insert_aux(root.left, value);
        root.left = balance(root, LEFT);
        o = root;
                                                                  root
                                                                                     0
      else {
        // Symmetrical
                                                                         h
                                                                  \nu \neq
                                                                 value
  return o;
```

```
Node* insert_aux(Node* root, int value) {
  Node* o;
  if (root == null) {
    o = new Node(value);
  else {
    if (root.value == value) {
      o = root;
    else {
      if (value < root.value) {</pre>
       root.left = insert_aux(root.left, value);
        root.left = balance(root, LEFT);
        o = root;
                                                                   root
                                                                                      0
      else {
        // Symmetrical
                                                                          h
                                                                   \nu >
                                                                  value
  return o;
```

```
Node* insert_aux(Node* root, int value) {
  Node* o;
  if (root == null) {
    o = new Node(value);
  else {
    if (root.value == value) {
      o = root;
    else {
      if (value < root.value) {</pre>
        root.left = insert_aux(root.left, value);
        root.left = balance(root, LEFT);
        o = root;
                                                                   root
                                                                                       0
      else {
        // Symmetrical
                                                                          h
                                                                   \nu >
                                                                   value
  return o;
                                               ins_aux
```