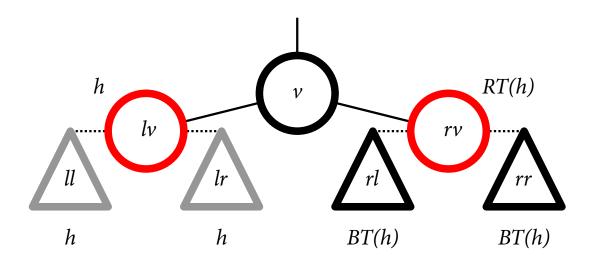
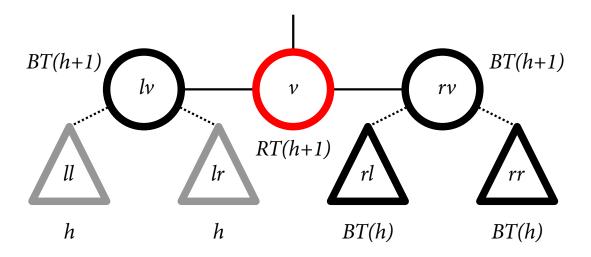
```
Node* balance(Node* root, int dir) {
if (red(root.left)) {
    if (red(root.right)) {
      root.left = blacken(root.left);
      root.right = blacken(root.right);
      root.black = false;
   else {
      if (red(root.left.left)) {
        root = rb.rotate.single(
          root, RIGHT);
      else {
        if (red(root.left.right)) {
          root = rb.rotate.dbl(
            root, RIGHT);
        else {
                                               h
                                                   lν
  else {
  return root;
```

```
Node* balance(Node* root, int dir) {
  if (red(root.left)) {
  if (red(root.right)) {
      root.left = blacken(root.left);
      root.right = blacken(root.right);
      root.black = false;
    else {
      if (red(root.left.left)) {
        root = rb.rotate.single(
          root, RIGHT);
      else {
        if (red(root.left.right)) {
          root = rb.rotate.dbl(
            root, RIGHT);
        else {
  else {
                                            h
                                                          h
                                                                        h
  return root;
```

```
Node* balance(Node* root, int dir) {
  if (red(root.left)) {
    if (red(root.right)) {
    root.left = blacken(root.left);
      root.right = blacken(root.right);
     root.black = false;
    else {
      if (red(root.left.left)) {
        root = rb.rotate.single(
          root, RIGHT);
      else {
        if (red(root.left.right)) {
          root = rb.rotate.dbl(
            root, RIGHT);
        else {
  else {
  return root;
```



```
Node* balance(Node* root, int dir) {
  if (red(root.left)) {
    if (red(root.right)) {
      root.left = blacken(root.left);
      root.right = blacken(root.right);
    _root.black = false;
    else {
      if (red(root.left.left)) {
        root = rb.rotate.single(
          root, RIGHT);
      else {
        if (red(root.left.right)) {
          root = rb.rotate.dbl(
            root, RIGHT);
        else {
  else {
  return root;
```



```
Node* balance(Node* root, int dir) {
  if (red(root.left)) {
    if (red(root.right)) {
      root.left = blacken(root.left);
      root.right = blacken(root.right);
      root.black = false;
    else {
    if (red(root.left.left)) {
        root = rb.rotate.single(
          root, RIGHT);
      else {
        if (red(root.left.right)) {
          root = rb.rotate.dbl(
            root, RIGHT);
       else {
  else {
                                            h
                                                          h
                                                                      BT(h)
  return root;
```

```
Node* balance(Node* root, int dir) {
  if (red(root.left)) {
    if (red(root.right)) {
      root.left = blacken(root.left);
      root.right = blacken(root.right);
      root.black = false;
   else {
      if (red(root.left.left)) {
      root = rb.rotate.single(
          root, RIGHT);
      else {
        if (red(root.left.right)) {
          root = rb.rotate.dbl(
            root, RIGHT);
                                                 RVT(h)
        else {
                              RT(h)
  else {
                             BT(h)
                                           BT(h)
                                                         BT(h)
                                                                       BT(h)
  return root;
```

```
Node* balance(Node* root, int dir) {
  if (red(root.left)) {
    if (red(root.right)) {
      root.left = blacken(root.left);
      root.right = blacken(root.right);
      root.black = false;
    else {
      if (red(root.left.left)) {
        root = rb.rotate.single(
          root, RIGHT);
      else {
        if (red(root.left.right)) {
          root = rb.rotate.dbl(
            root, RIGHT);
                                                        BT(h+1)
        else {
                              RT(h)
                                                                       RT(h)
                                                    lv
  else {
                                           BT(h)
                                                         BT(h)
                             BT(h)
                                                                        BT(h)
  return root;
```

```
Node* balance(Node* root, int dir) {
  if (red(root.left)) {
    if (red(root.right)) {
      root.left = blacken(root.left);
      root.right = blacken(root.right);
      root.black = false;
   else {
      if (red(root.left.left)) {
        root = rb.rotate.single(
          root, RIGHT);
      else {
      if (red(root.left.right)) {
          root = rb.rotate.dbl(
            root, RIGHT);
        else {
  else {
                                          BT(h)
                                                          h
                                                                       BT(h)
  return root;
```

```
Node* balance(Node* root, int dir) {
  if (red(root.left)) {
    if (red(root.right)) {
      root.left = blacken(root.left);
      root.right = blacken(root.right);
      root.black = false;
    else {
      if (red(root.left.left)) {
        root = rb.rotate.single(
          root, RIGHT);
      else {
        if (red(root.left.right)) {
         root = rb.rotate.dbl(
            root, RIGHT);
                                   RVT(h)
        else {
                                                        RT(h)
                                                    lrv
  else {
                                           BT(h)
                                                         BT(h)
                             BT(h)
                                                                       BT(h)
  return root;
```

```
Node* balance(Node* root, int dir) {
  if (red(root.left)) {
    if (red(root.right)) {
      root.left = blacken(root.left);
      root.right = blacken(root.right);
      root.black = false;
    else {
      if (red(root.left.left)) {
        root = rb.rotate.single(
          root, RIGHT);
      else {
        if (red(root.left.right)) {
          root = rb.rotate.dbl(
            root, RIGHT);
                                                        BT(h+1)
        else {
                              RT(h)
                                                                       RT(h)
                                                    lrv
  else {
                                           BT(h)
                                                         BT(h)
                             BT(h)
                                                                        BT(h)
  return root;
```

```
Node* balance(Node* root, int dir) {
  if (red(root.left)) {
    if (red(root.right)) {
      root.left = blacken(root.left);
      root.right = blacken(root.right);
      root.black = false;
    else {
      if (red(root.left.left)) {
        root = rb.rotate.single(
          root, RIGHT);
      else {
        if (red(root.left.right)) {
          root = rb.rotate.dbl(
            root, RIGHT);
        else {
                                             RT(h)
  else {
                                           BT(h)
                                                         BT(h)
                                                                       BT(h)
  return root;
```

```
Node* balance(Node* root, int dir) {
  if (red(root.left)) {
    if (red(root.right)) {
      root.left = blacken(root.left);
      root.right = blacken(root.right);
      root.black = false;
    else {
      if (red(root.left.left)) {
        root = rb.rotate.single(
          root, RIGHT);
      else {
        if (red(root.left.right)) {
          root = rb.rotate.dbl(
            root, RIGHT);
        else {
                                            BT(h)
                                                             h-1
                                         h-1
  else {
}
  return root;
```