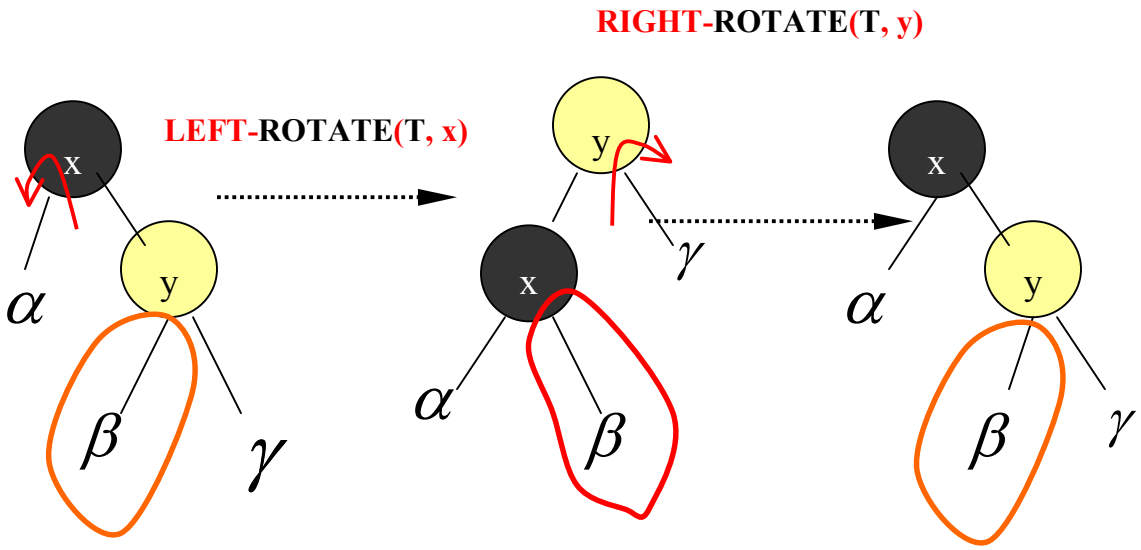
```
RB-INSERT(T, z)
{
      y = NIL;
      x = root[T];
      while(x <> NIL)
      {
            y = x;
            if(key[z] <key[x])
                  x = left[x];
            else
                  x = right[x];
      }
      P[z] = y;
      if( y == NIL)
            root[T] = z;
      else if (key[z] < key[x])
                  left[y] = z;
            else
                  right[y] = z;
      left[z] = NIL;
      right[z] = NIL;
      color[z] = "RED";
      RB-INSERT-FIXUP(T, z);
}
```

Search

if T is an empty tree.

Attach

ROTATION Concept: (α β γ are sub trees)



You can see the algorithm of LEFT-ROTATE(T, x) on **page 278**.

```
RB-INSERT-FIXUP(T, z)
{
      while(color[P[z]]==”RED”)
      {
            if(P[z] == left[P[P[z]])
            {
                  y=right[P[P[z]];
                  if(color[y]==”RED”)
                  {
                        color[P[z]] = “BLACK”;
                        color[y] = “BLACK”;
                        color[P[P[z]]] = “RED”;
                        z = P[P[z]];
                  }
                  else
                  {
                        if(z == right[P[z]])
                        {
                              z = P[z];
                              LEFT-ROTATE(T, z);
                        }
                        color[P[z]] = “BLACK”;
                        color[P[Pz]] = “RED”;
                        RIGHT-ROTATE(T,P[P[z]]);
                  }
            }
            else
            {
                  y=left[P[P[z]];
                  if(color[y]==”RED”)
                  {
                        color[P[z]] = “BLACK”;
                        color[y] = “BLACK”;
                        color[P[P[z]]] = “RED”;
                        z = P[P[z]];
                  }
                  else
                  {
                        if(z == left[P[z]])
                        {
                              z = P[z];
                              RIGHT-ROTATE(T, z);
                        }
                        color[P[z]] = “BLACK”;
                        color[P[Pz]] = “RED”;
                        LEFT-ROTATE(T,P[P[z]]);
                  }
            }
      }
      color[root[T]] = “BLACK”;
}
```

The uncle node is the same color as parent. So, change all parent’s uncle’s and grandparent’s colors.

Stretch **elbow** to a straight line.

Change colors and rotate a straight line into mountain shape.

The uncle node is same color as parent. So, change all parent’s uncle’s and grandparent’s colors.

Stretch **elbow** to a straight line.

Change colors and rotate a straight line into mountain shape.