RBT Insertion

```
Insert(key)
if tree is empty
      insert key as black root
else
      BST insert key as red leaf
      FixRBT at newly inserted leaf

FixRBT(N)
if P is black
      do nothing
else if U is red
      change P and U to black
      if G is root
            do nothing
      else
            change G to red
            FixRBT(G)
else if G, P and N form a straight line
      swap color of G and P
      rotate along G such that P replaces G
else
      rotate along P such that G, P and N a straight line
      FixRBT(N)
```

RBT Deletion

```
Delete(N)
follow BST deletion procedure and ensure that N is leaf node
if N is red
      simply delete
else
      replace N with DB nil node
      ResolveDB at N

ResolveDB(N)
if N is root
      remove DB
else if S is red
      swap color of S and P
      rotate along P such that S replaces P
      ResolveDB at N
else if children of S are black
      change S to red
      remove DB
      if P is black
            change P to DB
            ResolveDB at P
      else
            change P to black
else if far child of S is red
      swap color of P and S
      rotate along P such that S replaces G
      change far child to black
      remove DB
else
      swap color of near child and S
      rotate along S such that S becomes far children
      ResolveDB at N
```