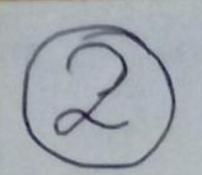To insert a Node k into a red-black tree T

Case 1 :- T is empty
        make k the root node and make its colour
        black.

Case 2 :- P is black   ~ parent node of k
        if P is black, none of the properties are
        violated, nothing must be done

Case 3 :- P is ~~read~~ red { adjacent nodes are red }

→ if u { P → is also red }
change colour of P and u to black
change colour of grandparent to red, (if it is
not the root node).

→ if u {&P } is black or NULL.

a)   P is right of G & k is right of P
    perform left rotation at G and interchange
color of G & P.

b) ~~p~~ P is right of G & k is left of P, right
    rotate P and it reduces to 'ⓐ'.

c) P is left of G & k is left of P, right rotation
    at G & interchange ~~color~~ colour of G and P.

d) P is left of G & k is right of P, left
    rotation at P and it reduces to ⓒ.

Rutazeet.

```
insertion (T, k){
    bst insertion (T, k)
    while k. parent. colour == RED
        if  k. parent == k.parent.parent.right.
        u = k. parent. parent. left.
        if      u.colour == RED
                u. colour = BLACK
                k.parent. colour = BLACK
                k. parent. parent. colour = RED
                k = k. parent. parent
        else if  k == k. parent. left
                k = k. parent
                left rotate (T, k)
                k.parent. colour = BLACK
                k. parent. parent. colour = RED
                right rotate (T, k. parent. parent)
        else (same as then with left, right
                exchanged).
    T. root. colour = BLACK.
```