

SPLAY TREES - Loosely balanced BST

Insert/Search - Promote the element inserted/
Searched as the root node

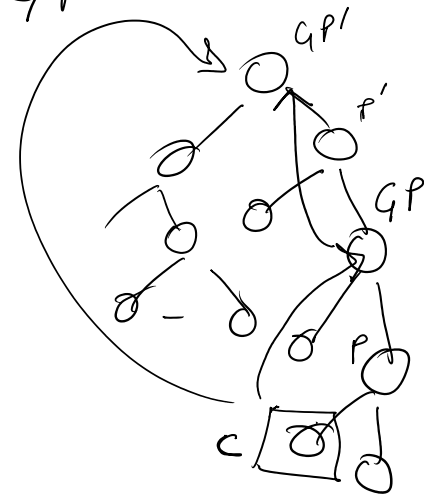
Delete - Promote the node whose child
has been deleted to the root posⁿ

SPLAYING / SPLAY A NODE

↳ Make / propagate that node
to the root position

Promote a node to its GP

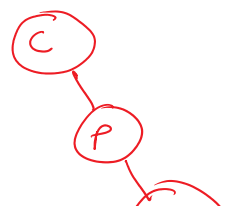
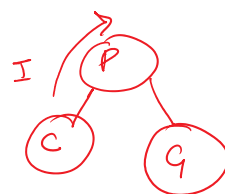
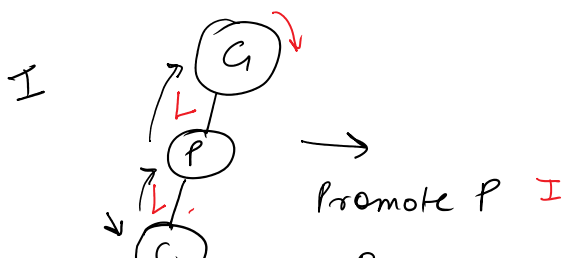
C P GP
Child Parent Grand parent

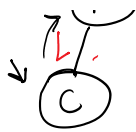


1 small step

Node will be promoted to GP

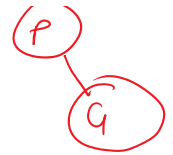
Orientations with respect to the GP





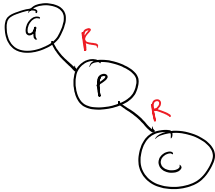
Promote P I

Promote C II



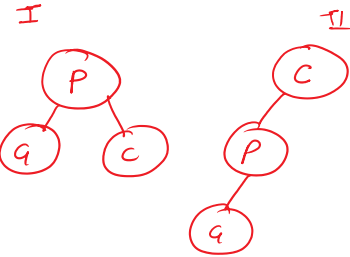
LL rotation

II



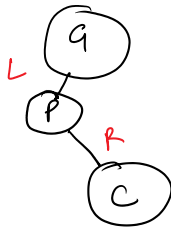
Promote P I

Promote C II



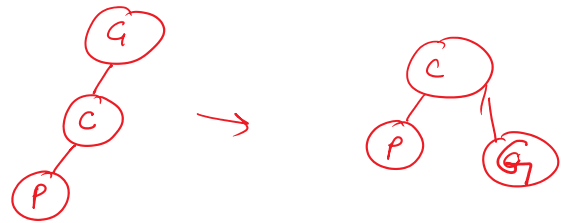
RR rotation

III



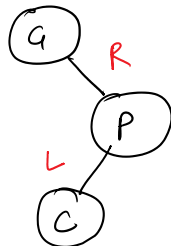
Promote C I

Promote C II



LR rotation

IV



Promote C

Promote C

V

Special case

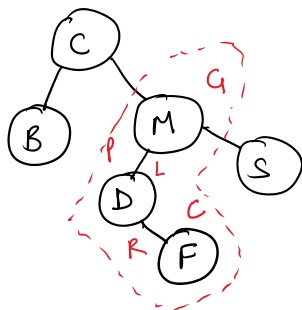


Promote C

RL rotation

* Promote a node = Rotate its parent

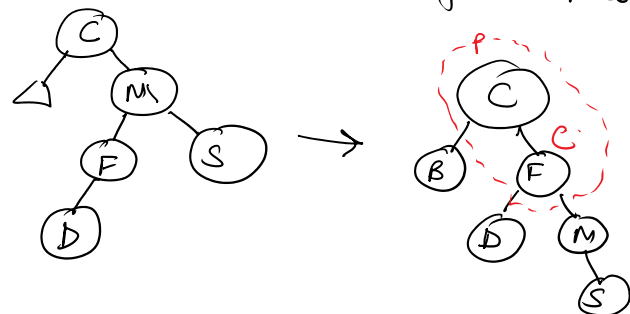
#g.



case II

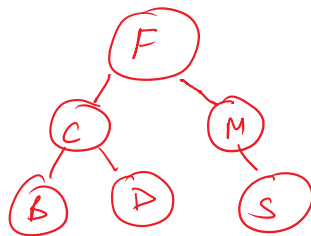
Splay node F \Rightarrow make F the root of the tree

Promote F
Promote F



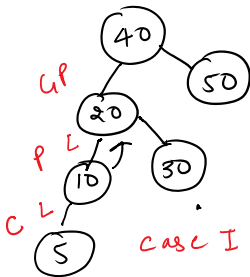
case I

5

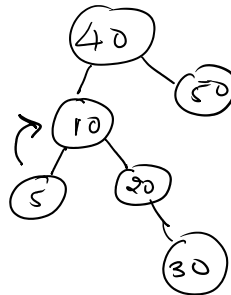


← Case IV
Promote F

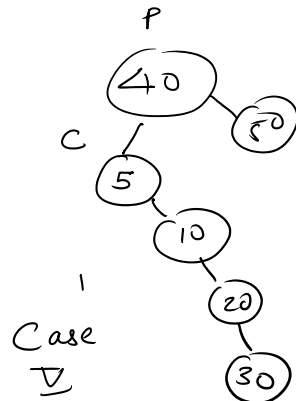
#9



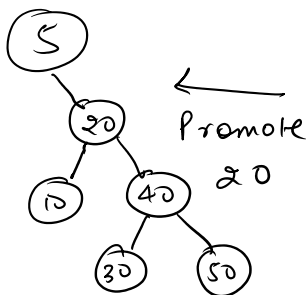
Promote 10



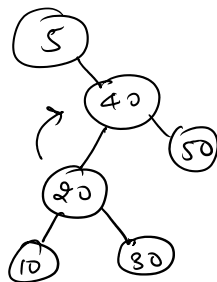
→ Promote 5



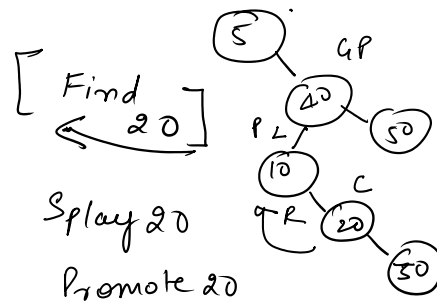
Promote 5



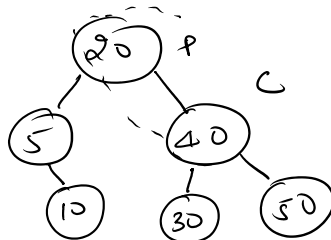
← Promote 20



Case IV



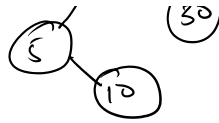
Promote 20
Case V



"Splay 40"
Splay the parent
of the node which
was deleted

Promote 40





* Not balanced tree

* Locality of reference

* Mem overhead vs AVL trees → Stack