

Example similar to homework

inputs in order: D, F, X, T, B, Q, G, Y, C, Z, S, A

If the new input coming in is larger, it goes to the right, if it is smaller, it goes to the left.

input D:

D

input F:

D → F

input X:

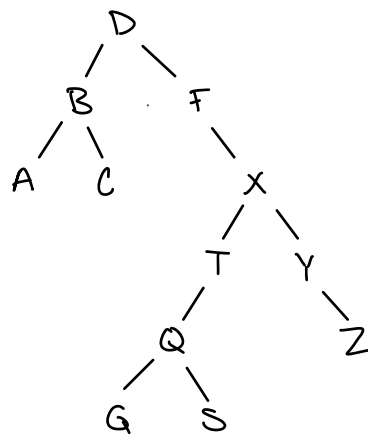
D → F → X

input T:

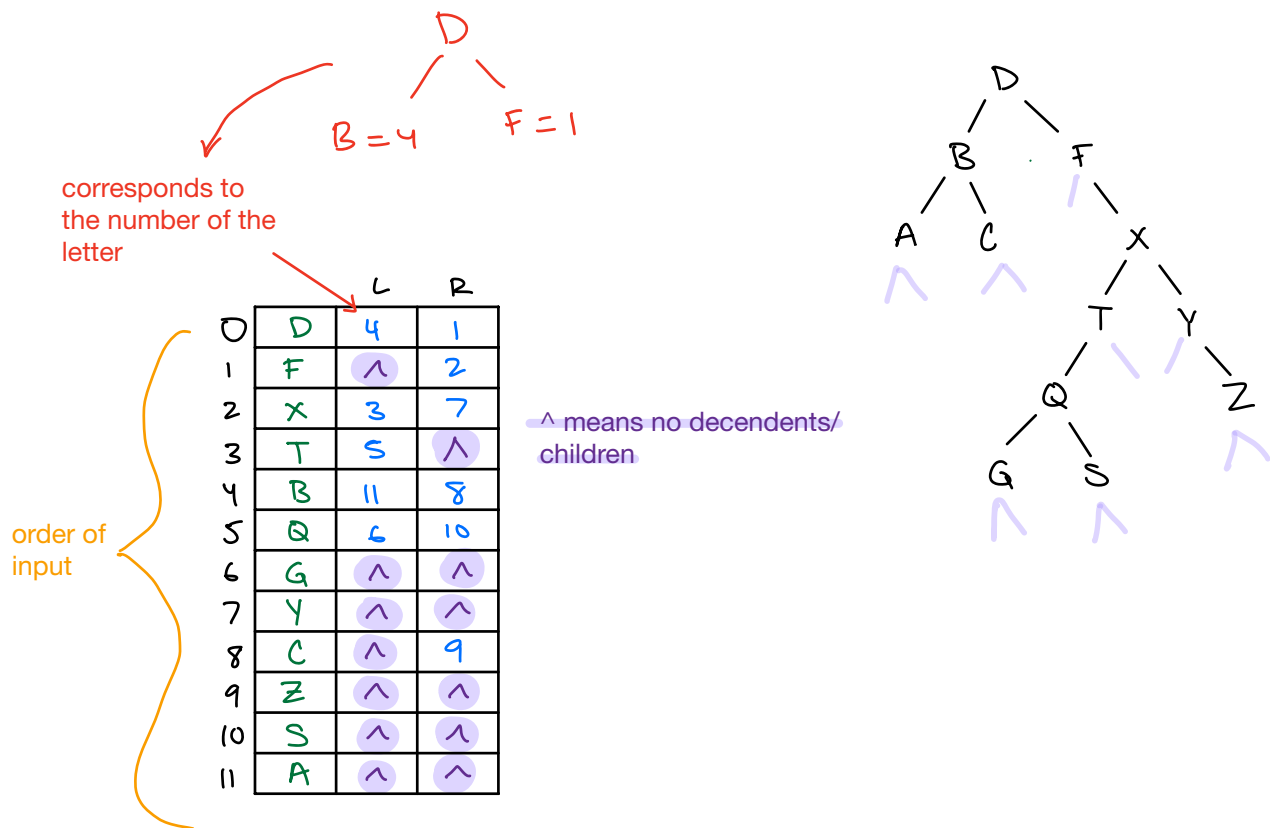
D → F → X
↓
T

T is greater than D,
greater than F but
less than X so it goes
down to the left

Resulting Binary Search Tree



How to store the Binary Search Tree using Key, Left and Right from previous problem

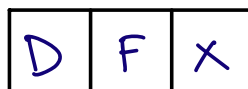


Sorting a Binary Search Tree of order 3 using previous inputs

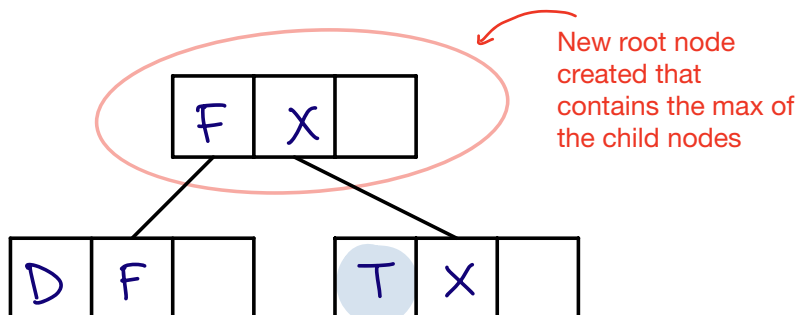
**order 3 means every node can hold 3 inputs

inputs in order: D, F, X, T, B, Q, G, Y, C, Z, S, A

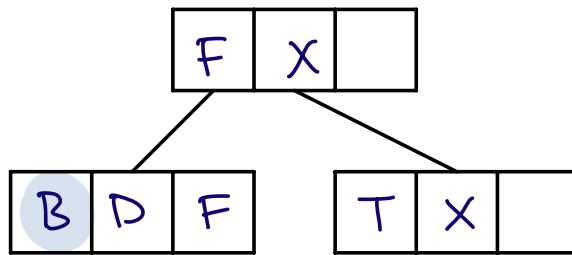
inserting D, F, X is straight forward because a single node can hold the 3 inputs



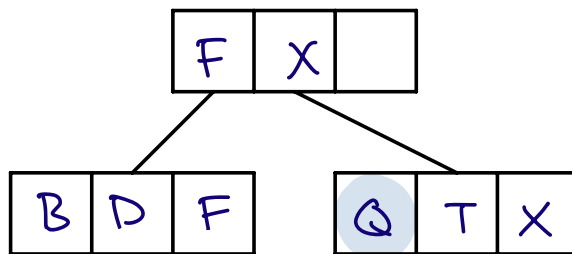
Insert T: T does not fit so the node is split, and T is placed before X



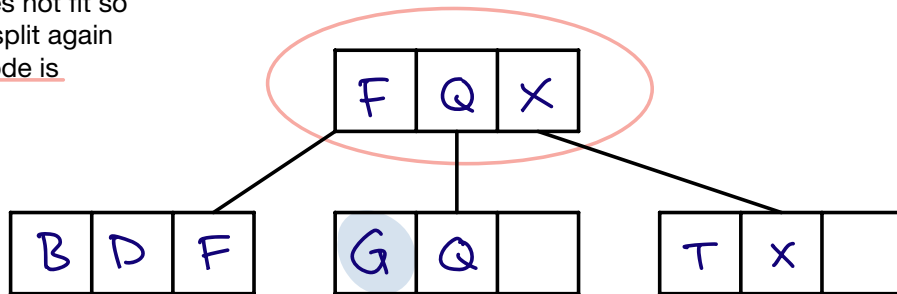
Insert B:



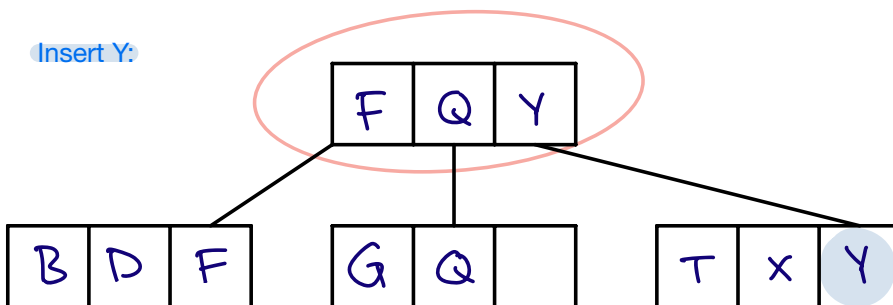
Insert Q:



Insert G: G does not fit so
the nodes get split again
and the root node is
changed

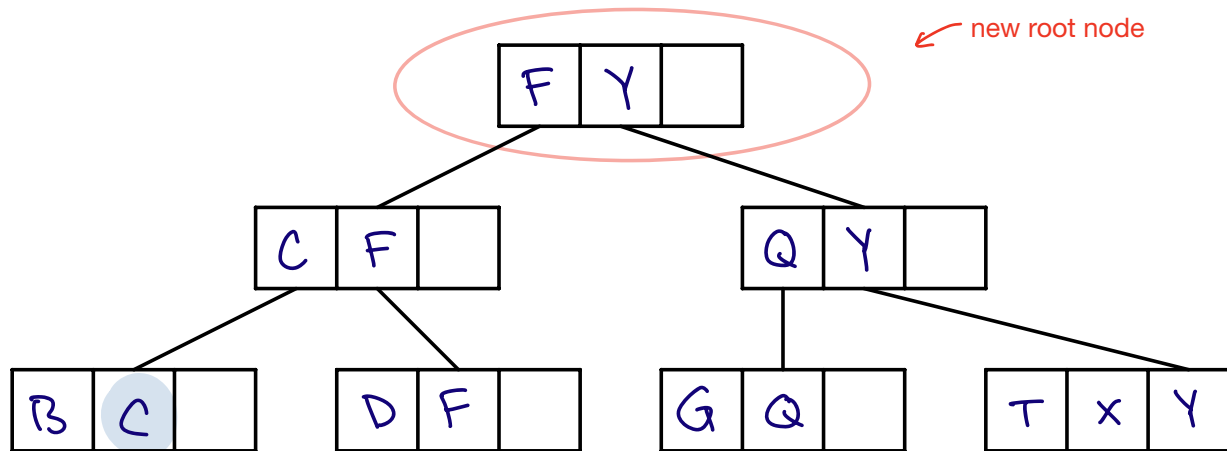


Insert Y:

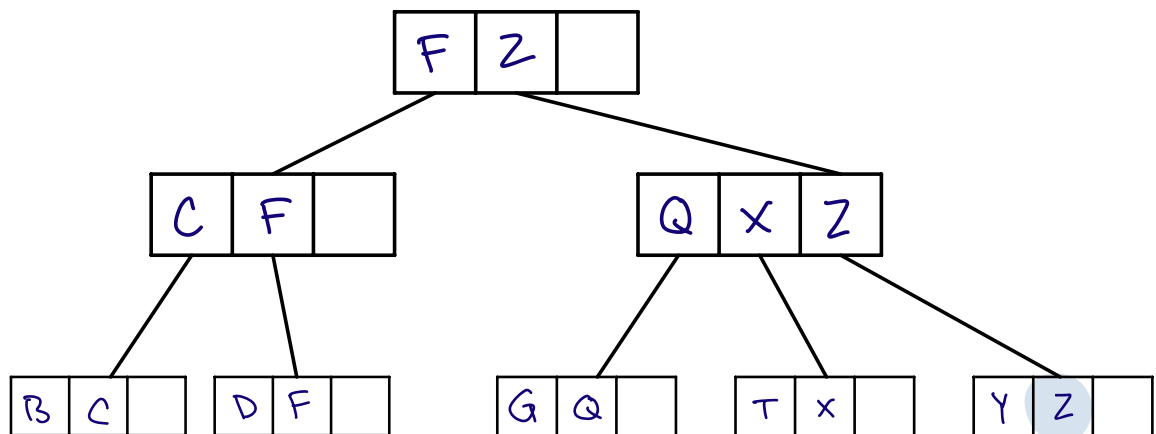


Y is now the max so
root changes

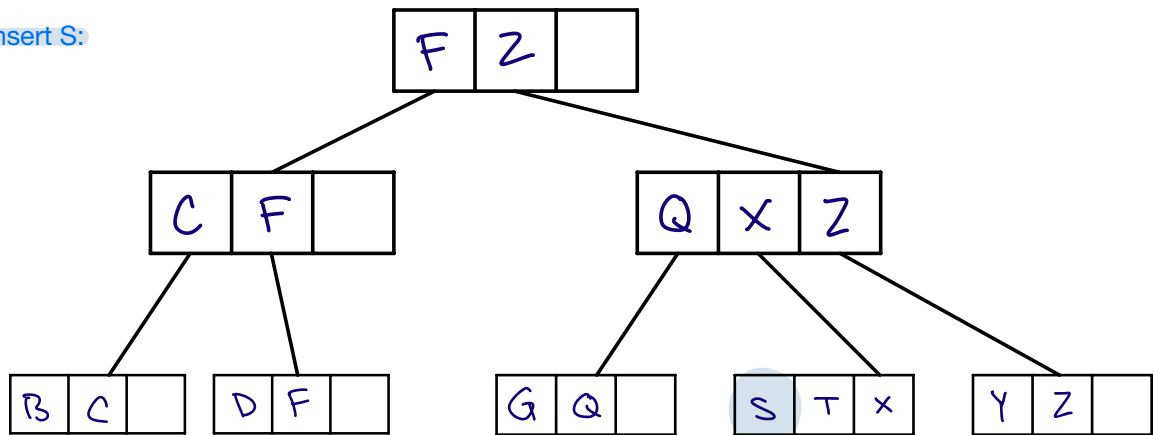
Insert C: C does not fit so the nodes are split again BUT now there are 4 max nodes so the root node needs to be split and a new root is created



Insert Z: split node remember to update root



Insert S:



Insert A:

