

POTD 8

Aleksander Schultz (aps4yj), Justin Kaplan(jrk5ak), Kavın Bapat (kb7pcr)

1.

Start by pushing 10, 15, 20, and 30 to node. 60 would fill node up, so split down the middle, using 20 as a key, and continue adding. Now have two nodes:

10	15		
----	----	--	--

and

20	30	60	65
----	----	----	----

And the tree root is:

20			
----	--	--	--

Then try to insert 80. Right node is full so split on 60, to create three nodes.

10	15		
----	----	--	--

and

20	30		
----	----	--	--

and

60	65	80	95
----	----	----	----

And the tree root is:

20	60		
----	----	--	--

Then, we try to add 100, but node 3 is full. Split on 80.

Nodes are now:

10	15		
----	----	--	--

and

20	30		
----	----	--	--

and

60	65		
----	----	--	--

and

80	95	100	112
----	----	-----	-----

And the tree root is:

20	60	80	
----	----	----	--

Try to insert 124 and fail. Split on 100.

Nodes are now:

10	15		
----	----	--	--

and

20	30		
----	----	--	--

and

60	65		
----	----	--	--

and

80	95		
----	----	--	--

and

100	112	124	140
-----	-----	-----	-----

And the tree root is:

20	60	80	100
----	----	----	-----

Finally, try to insert 148. Rightmost node is full, so split last node into

100	112		
-----	-----	--	--

and

124	140	148	
-----	-----	-----	--

However, parent is also full. Thus, need to split the parent.

Parent becomes

20	60		
----	----	--	--

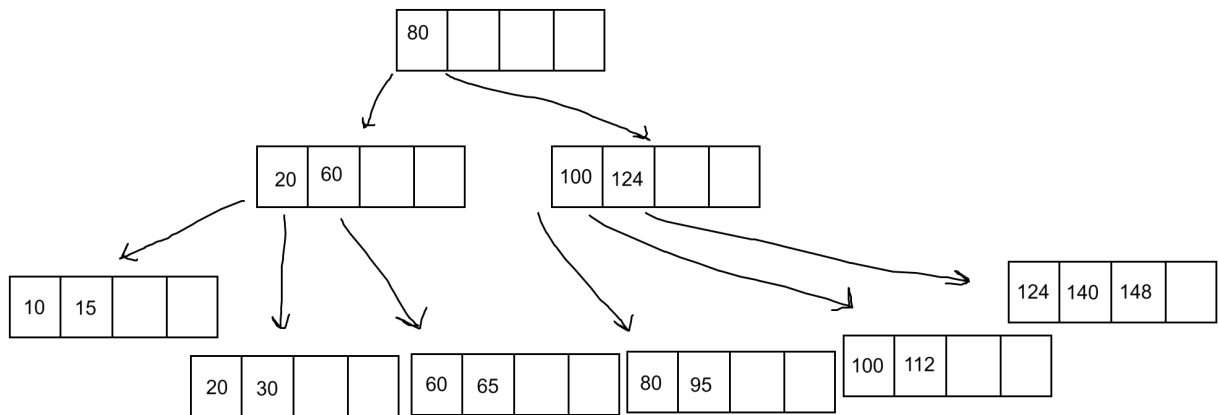
and

100	124		
-----	-----	--	--

Because parent is root node, need to go a level up to create new root:

80			
----	--	--	--

Thus, final tree is:

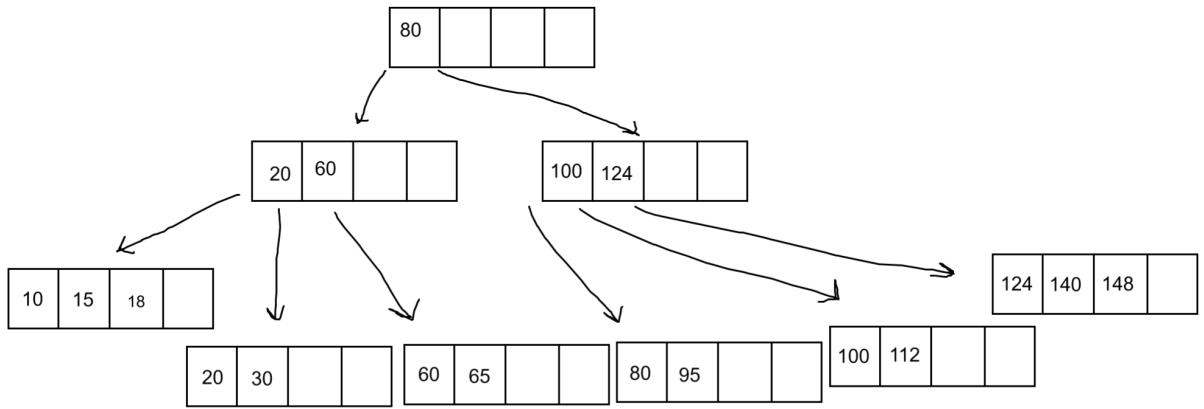


2. Use the B+ tree constructed in question 1, what is the **minimum** number of pointers to be followed to satisfy the query: Get all records of data associated with key(s) between 40 and 85?

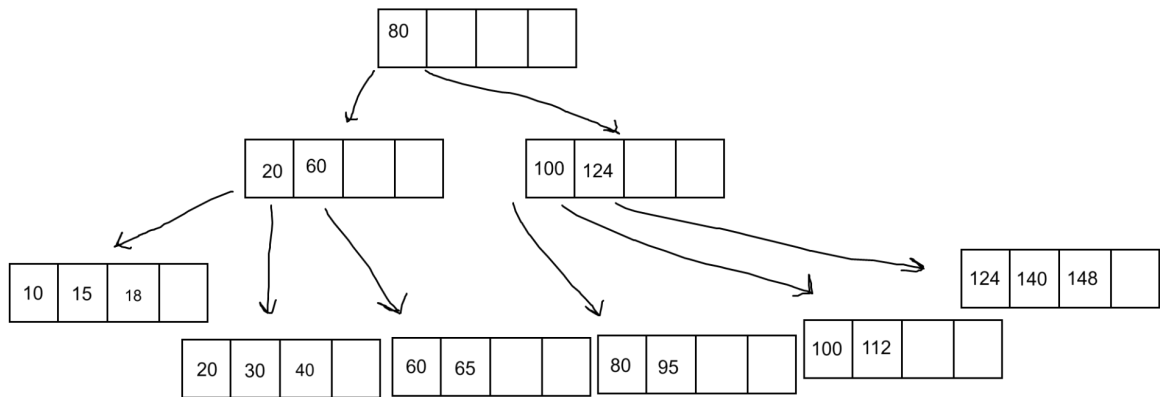
Need at least **4 pointers**. Go left at root node, go down to second bottom level node (20, 30), go laterally to third bottom level node (60, 65), go laterally to fourth bottom level node (80, 95). 85 ends between the values there so we're done at 4 pointers.

3. Use the B+ tree from question 1, apply each of the following operations in order, and show the tree after each operation

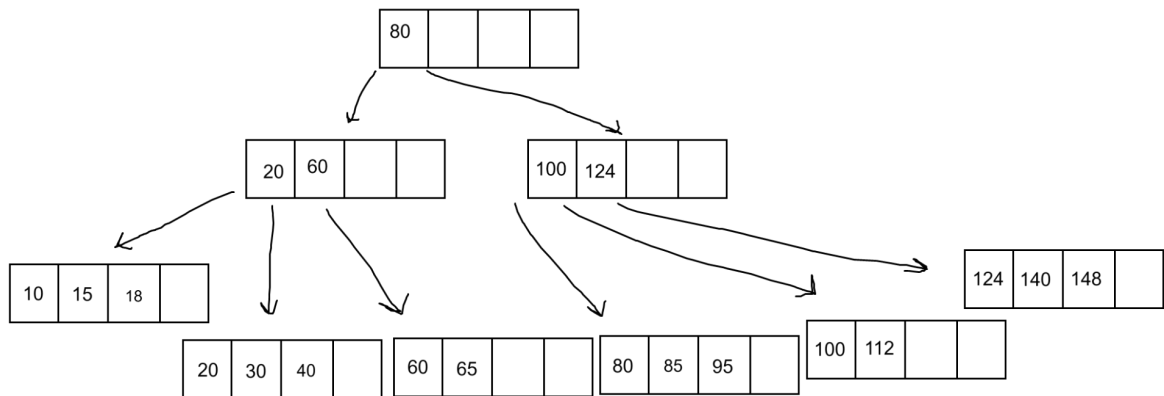
3.1 Insert 18



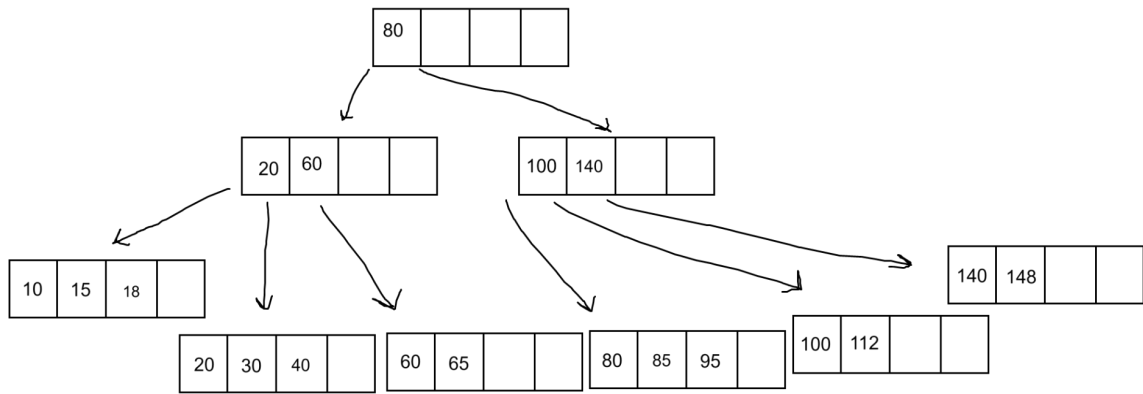
3.2 Insert 40



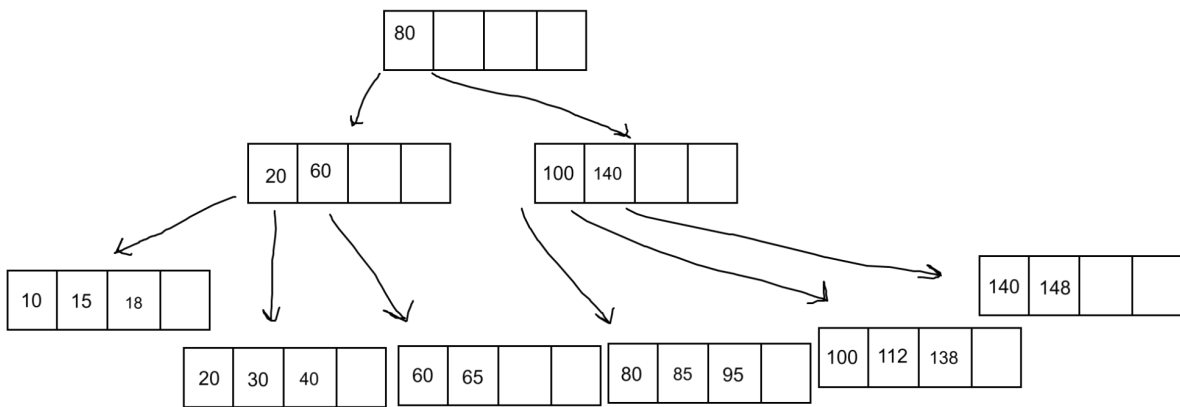
3.3 Insert 85



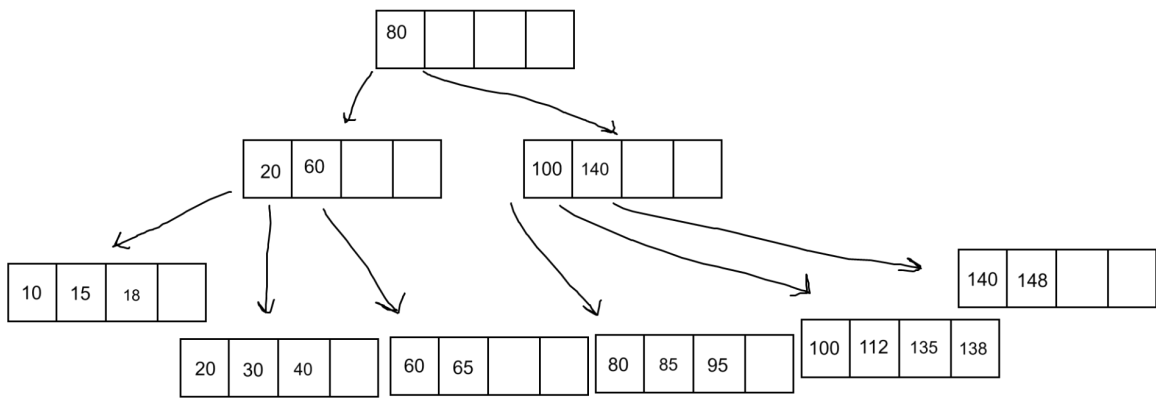
3.4 Delete 124



3.5 Insert 138



3.6 Insert 135



3.7 Insert 120

