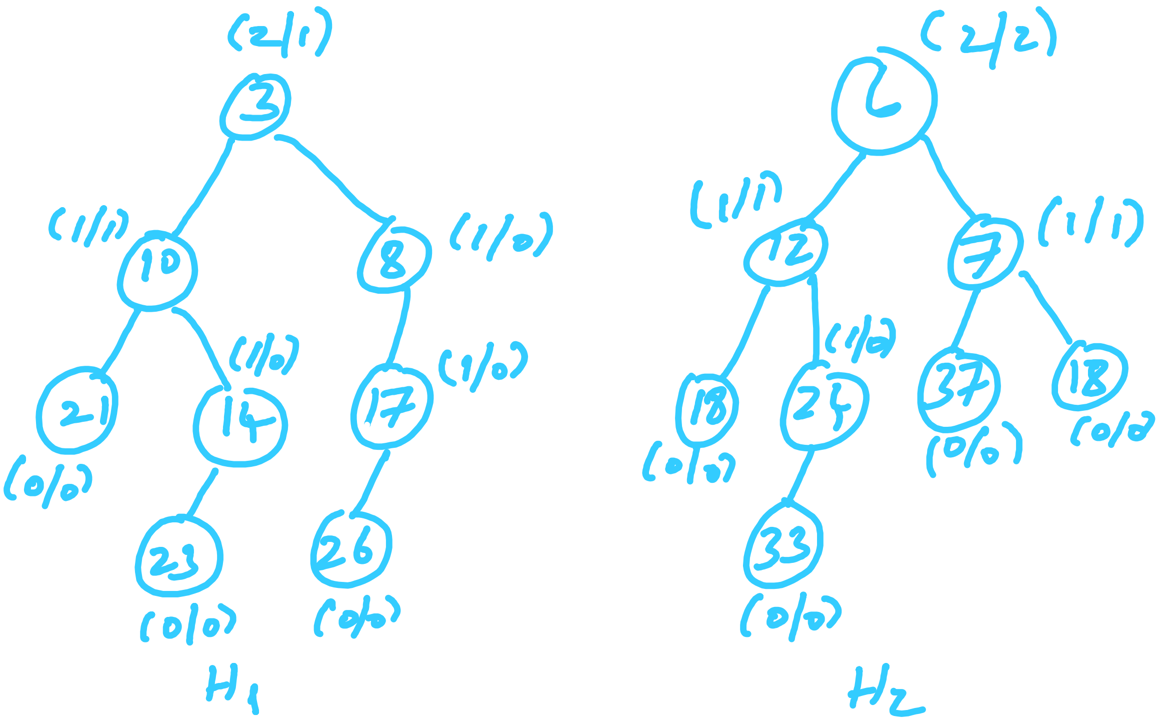
Leftist heap merge example

16 November 2020 19:22

Consider the following two heaps H_1 & H_2.



Merge H_1 & H_2

Step 1:

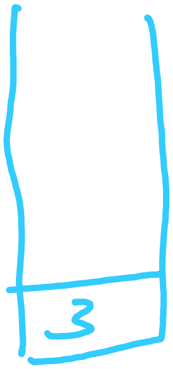
Compare roots of H_1 & H_2.

Root $(H_1) = 3$

Root $(H_2) = 6$

Minimum is 3.

Push to stack



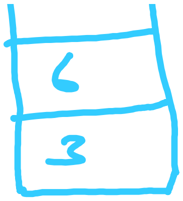
Step 2:

Compare root of right subtree of 3 with root of H_2.

(ie) 8 and 6.

Min is 6.

Push to stack.



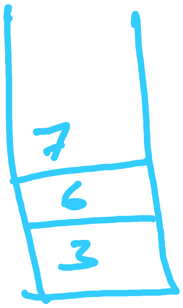
Step 3:

Compare root of right subtree of 6 with 8

(ie) 7 with 8.

Min is 7

Push to stack



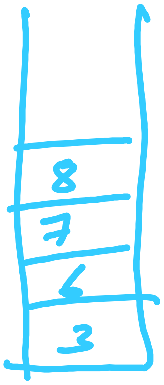
Step 4:

Compare root of right subtree of 7 with 8.

(ie) 18 with 8.

Min is 8.

Push to stack

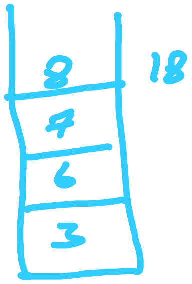


Step 5

Compare root of right subtree of 8 with 18.

(ie) Null with 18.

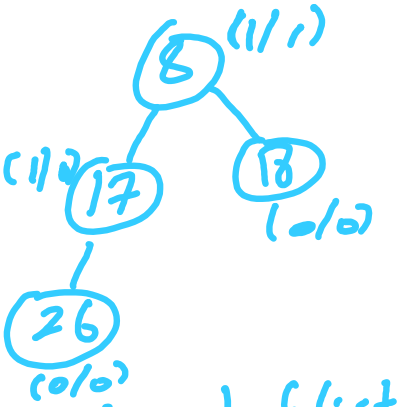
Since it is null, return other argument (ie) 18



Step 6

Step :

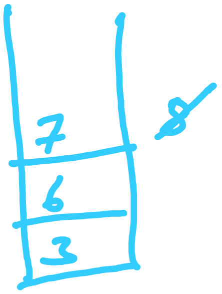
Merge Subtree with root 18 to right of Subtree with root 8.



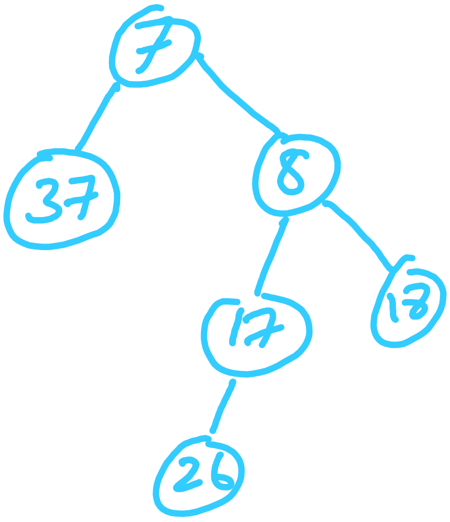
Check leftist.
Satisfied.

Step 7

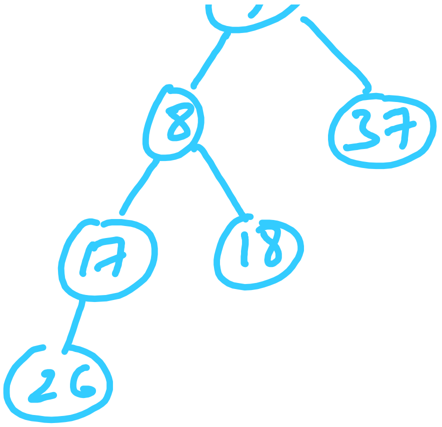
Return 8 from Stack



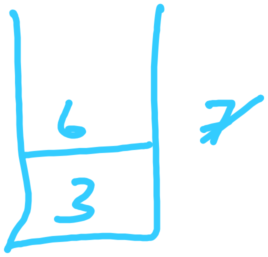
Attack Subtree with root 8 to right of 7.



Check Leftist. Not Satisfied.
Hence swap Children.

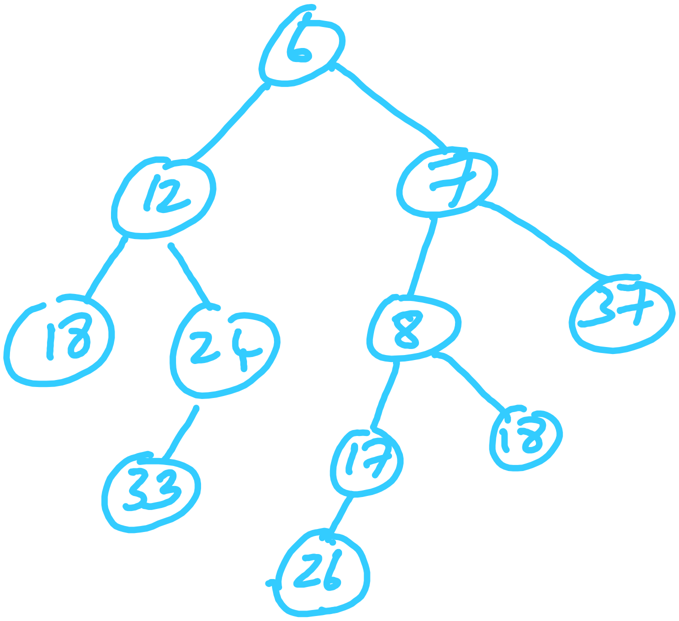


Step 8 .



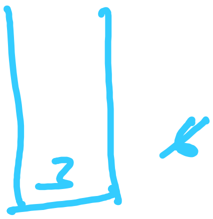
Return 7

Attach Subtree with root 7 to right of 6.



Check Leftist.
Satisfied.

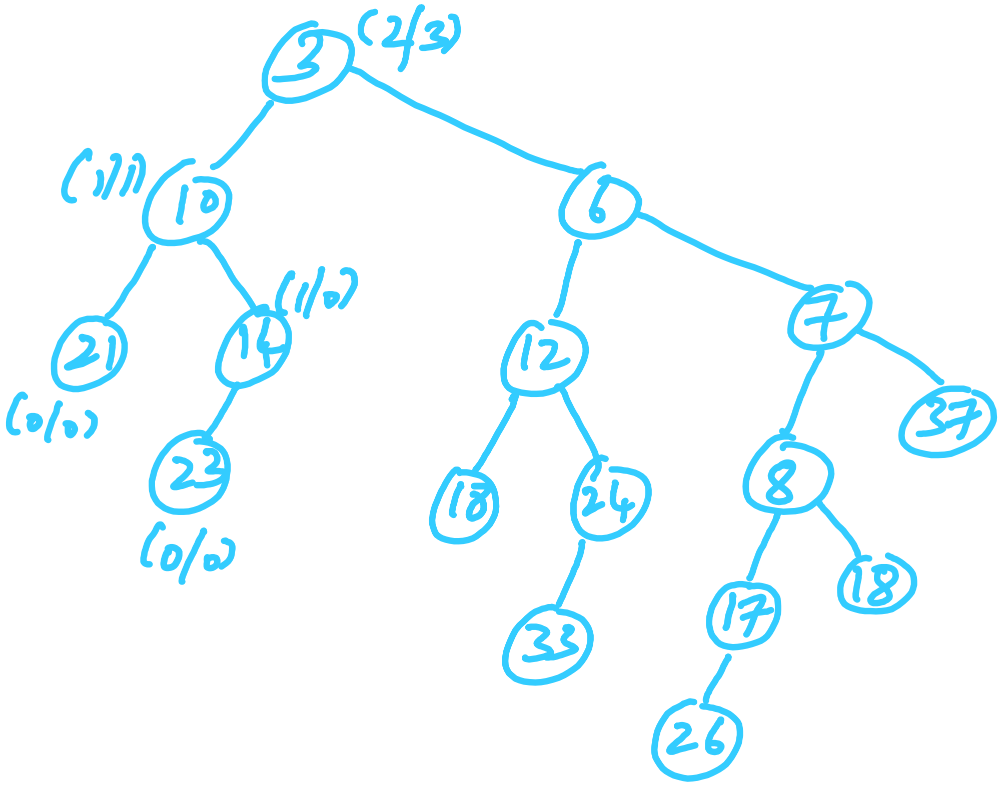
Step 9



Return 6.

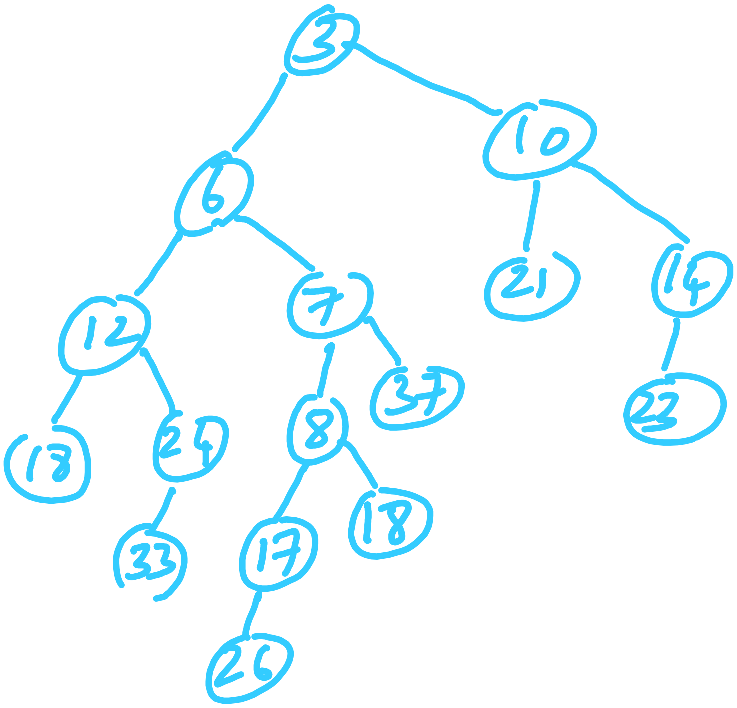
Attach Subtree with root 6

to righ of 3.



Not satisfied.

∴ Swap children.



Step 10

Return 3. Stack empty

Stop.

∴ Final merge tree is shown above.