School of Electrical and Computer Engineering

CptS223: Advanced Data Structures C/C++

Spring 2018

Homework #6

Due: Friday 03/23/2018 @ 11:59pm

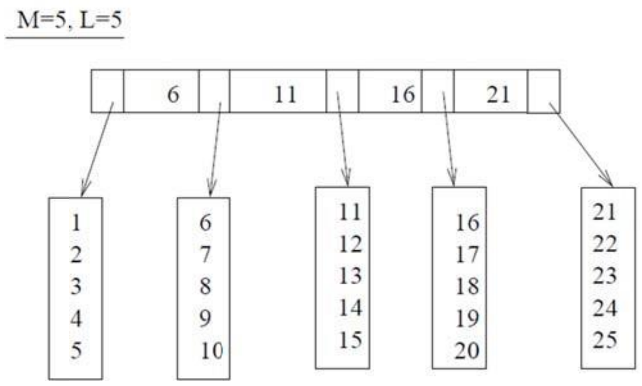
Rules/In structions: This is an individual assignment and all work must be your own. Be sure to show your work when appropriate. This assignment must be turned in by the due date via Blackboard in *PDF* format. You may use any editor you like (or even print it out, *legibly* write in answers, and scan it in), but convert it to *PDF* for submission.

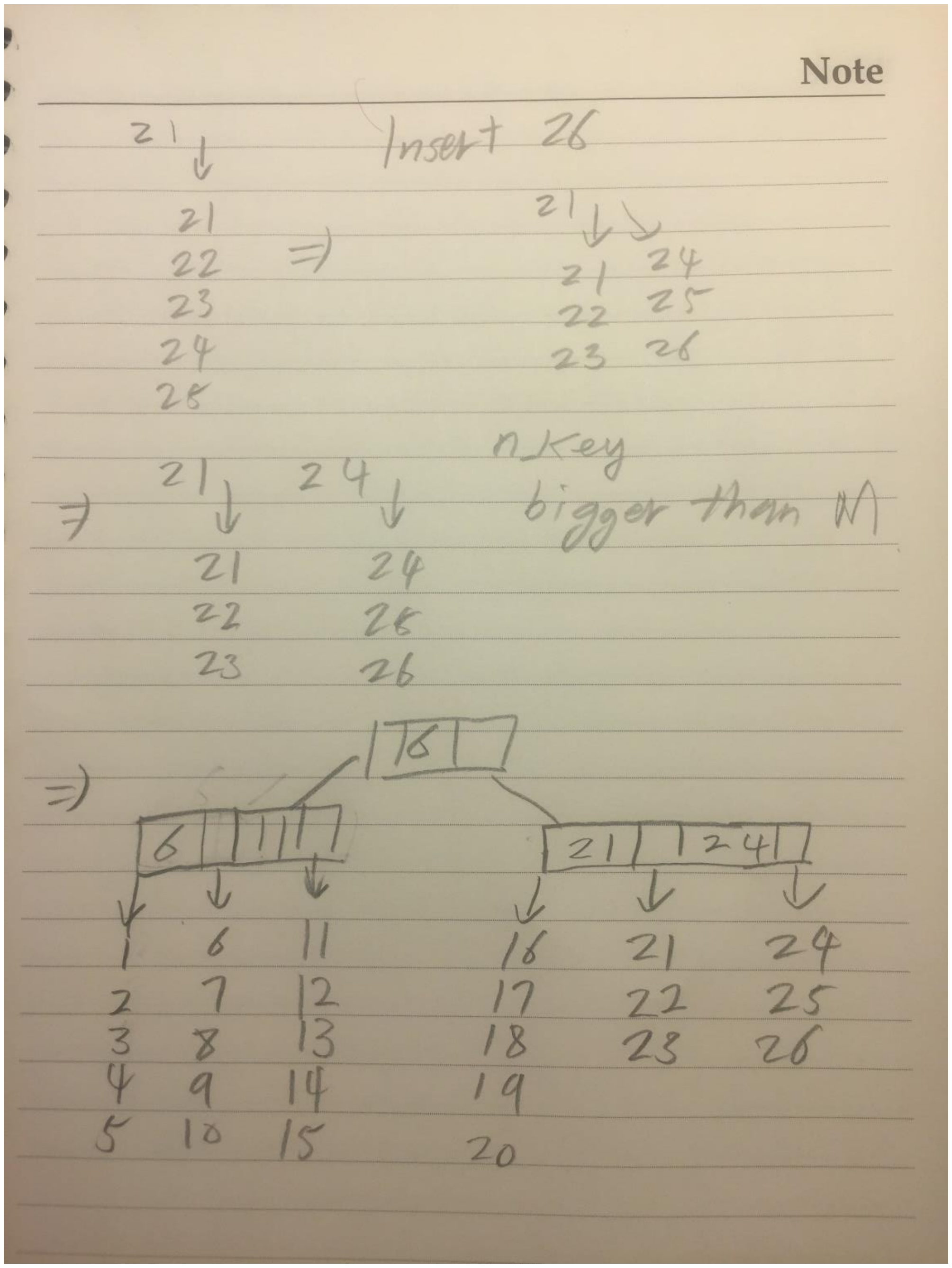
A maximum of 24 hours delay is permitted at the cost of 10% penalty. Beyond this 24-hour delay, not submissions will be graded. That is, assignments submitted after 24 hours after the above due date will not get any credits.

Grading:

- ✓ 10 points for Problems 1, 2, 3, 6
- ✓ 15 points for Problems 4, 5, 7, 8

1. Insert 26 into the following B+ tree:



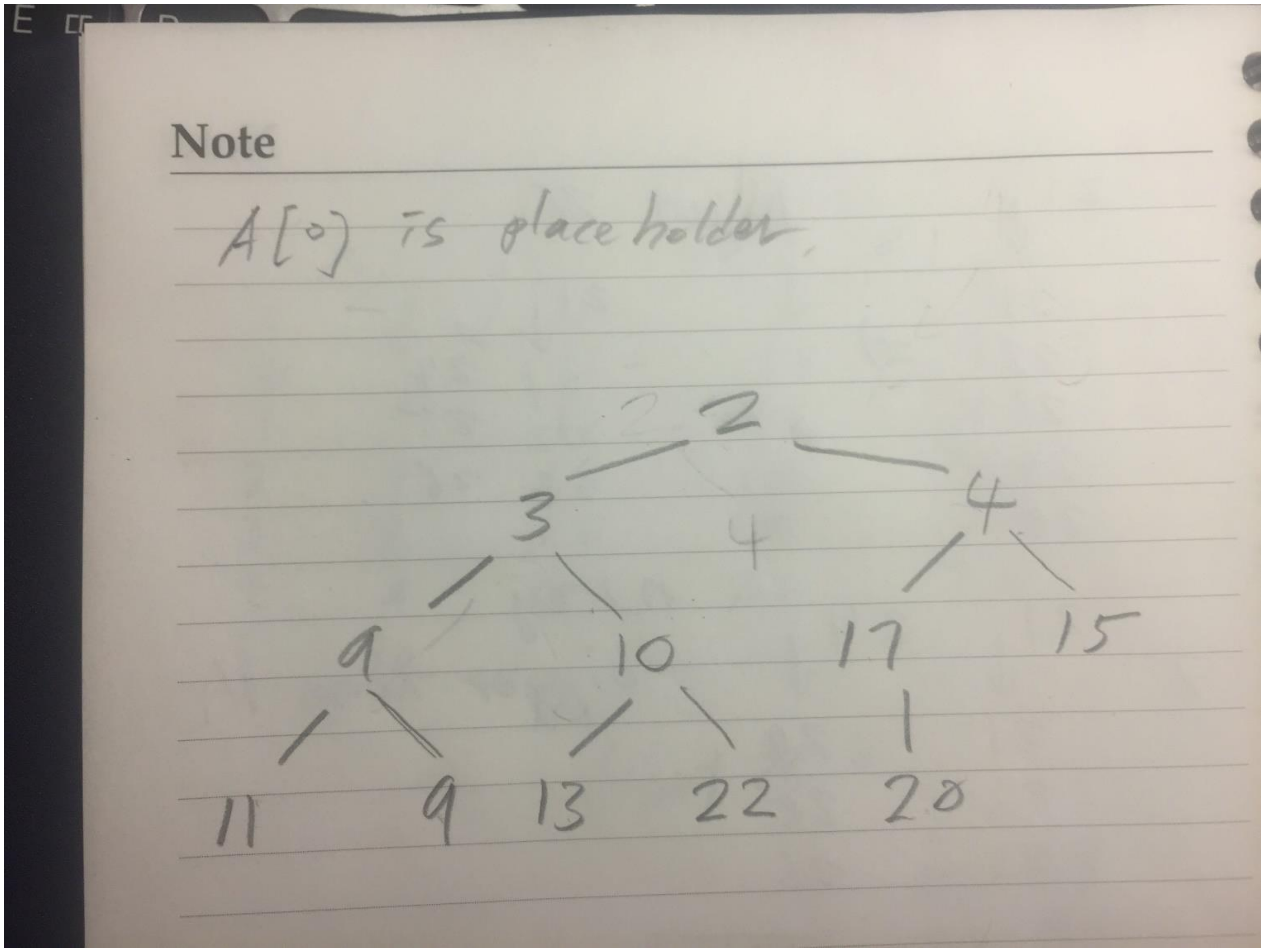


2. Calculate the parameters M and L for a B+Tree that meets the following specifications: Each data record in the array is 32 bytes. The search key occupies 12 bytes. Each disk block is 4 KB (=4,096 bytes). Assume 64-bit CPU architecture (i.e., pointers cost 8 bytes each). The data occupation must be smaller than assigned disk block.

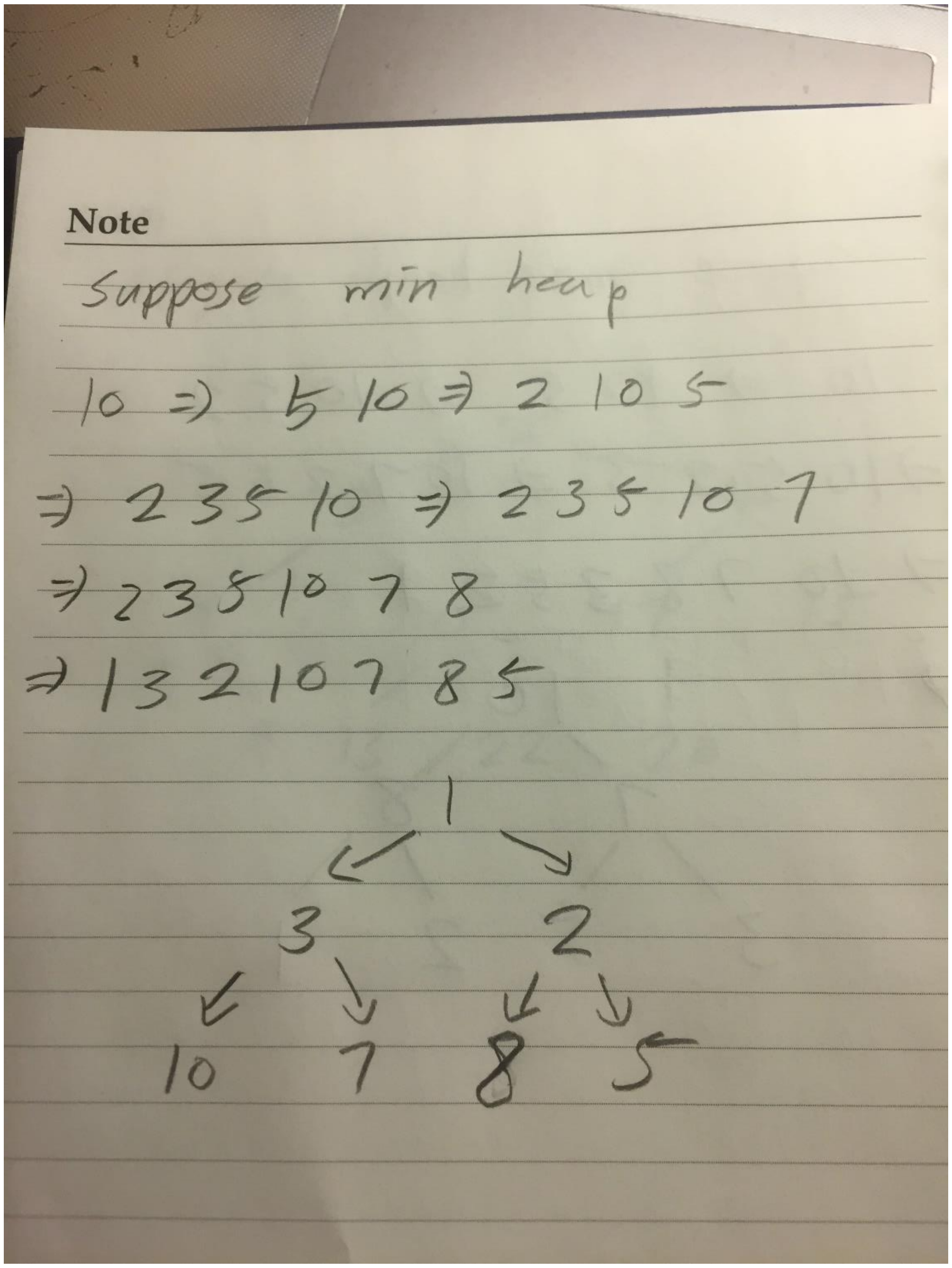
```
So, 4096 >= n*complete block.
4096 >= (n_pointer+n_key)(b_pointer+b_key) + (b_pointer)
4096 >= M(8+32) + 8
M <= 102.2
Maximum M = 102
Maximum L = M+1 = 103
```

3. Draw the binary heap in its tree form from its array form:

0	1	2	3	4	5	6	7	8	9	10	11	12
	2	3	4	9	10	17	15	11	9	13	22	20



4. Starting with an empty binary heap, insert the following sequence of elements into it: 10, 5, 2, 3, 7, 8, 1. Your answer should show the resulting binary heap after each insertion. However, there is no need to show the intermediate trees *within* an insertion step.



Note

suppose min heap

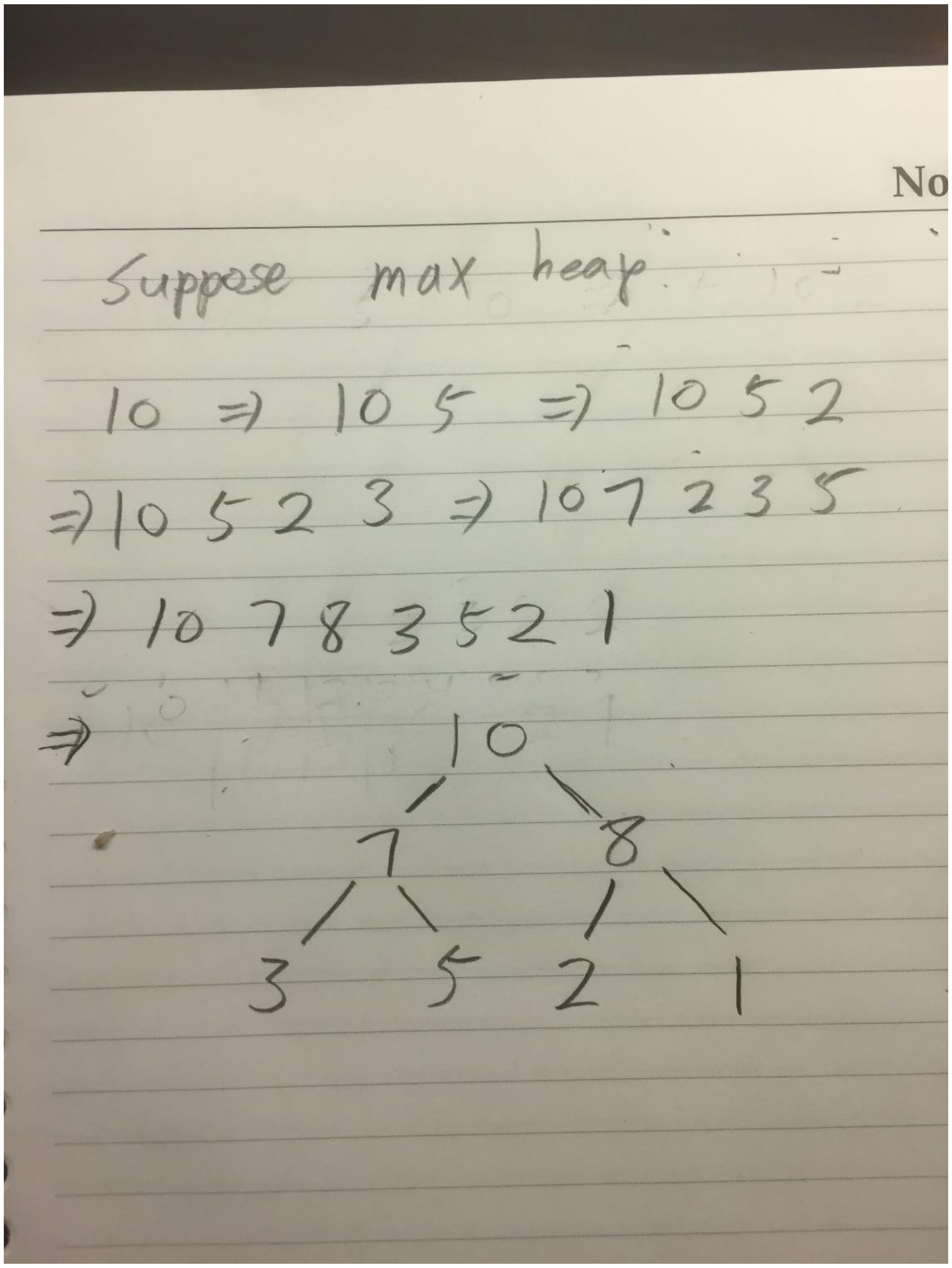
10 ⇒ 5 10 ⇒ 2 10 5

⇒ 2 3 5 10 ⇒ 2 3 5 10 7

⇒ 2 3 5 10 7 8

⇒ 1 3 2 10 7 8 5

$$\begin{tikzcd}
 & & & 1 \arrow[dll] \arrow[drr] & & & \\
 & 3 \arrow[dl] \arrow[dr] & & & & 2 \arrow[dl] \arrow[dr] & \\
10 & & 7 & & 8 & & 5
\end{tikzcd}$$


No

Suppose max heap

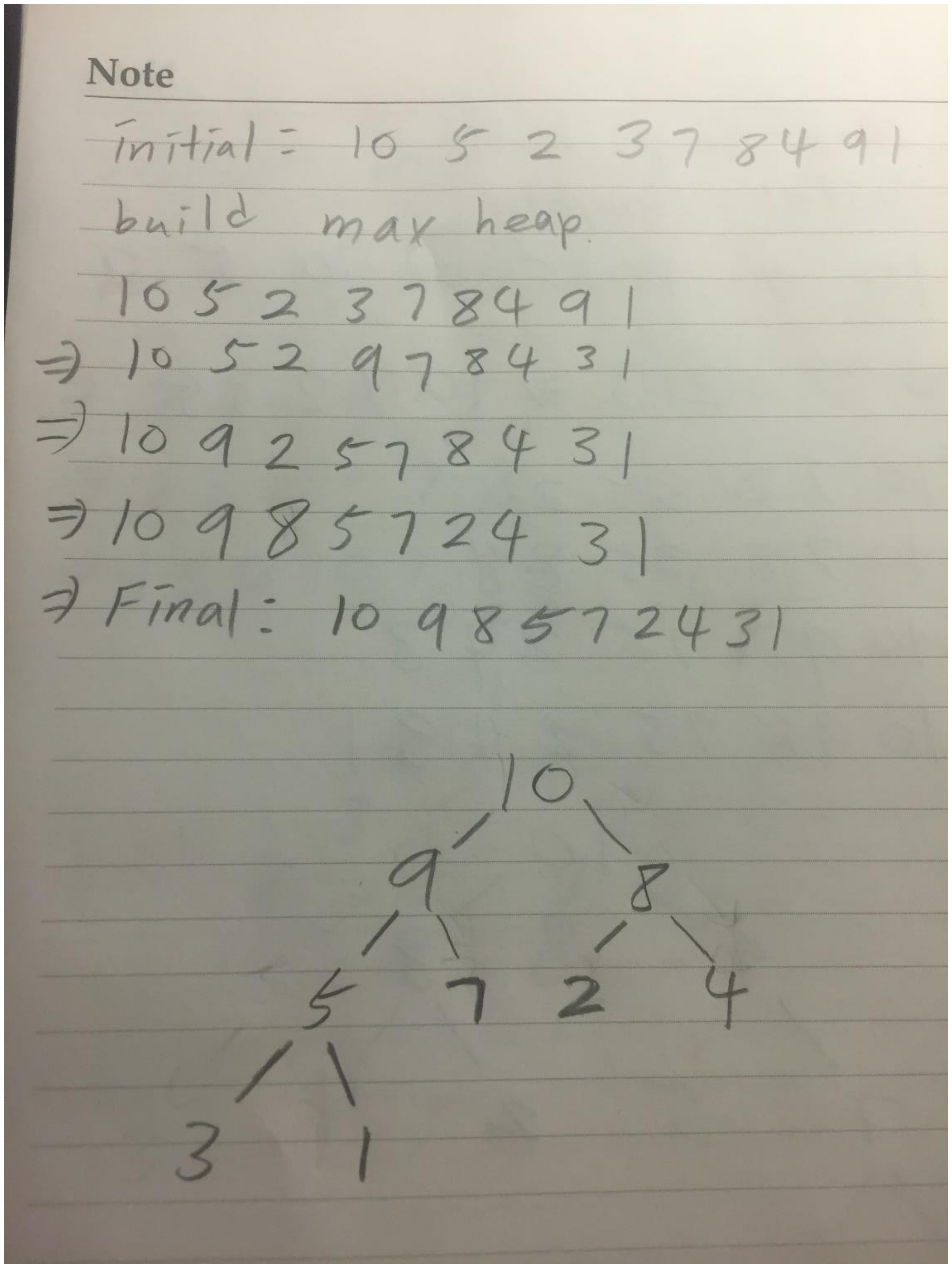
10 ⇒ 10 5 ⇒ 10 5 2

⇒ 10 5 2 3 ⇒ 10 7 2 3 5

⇒ 10 7 8 3 5 2 1

⇒

5. Build a heap for the following set of elements using the BuildHeap() method: {10,5,2,3,7,8,4,9,1}. You can populate the initial heap for the BuildHeap() function in any order of elements you want. But make sure you show the heap at different steps of the BuildHeap function, starting from your initial heap to your final binary heap.



Note

initial = 10 5 2 3 7 8 4 9 1

build max heap

10 5 2 3 7 8 4 9 1

⇒ 10 5 2 9 7 8 4 3 1

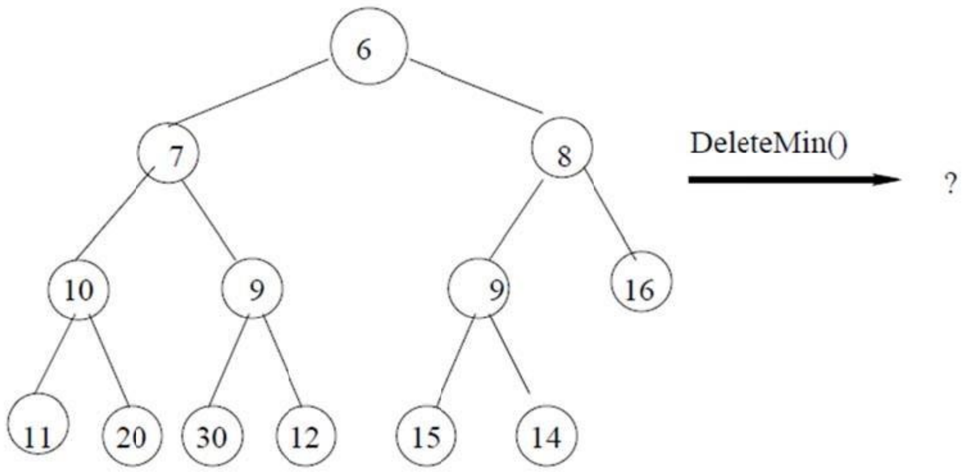
⇒ 10 9 2 5 7 8 4 3 1

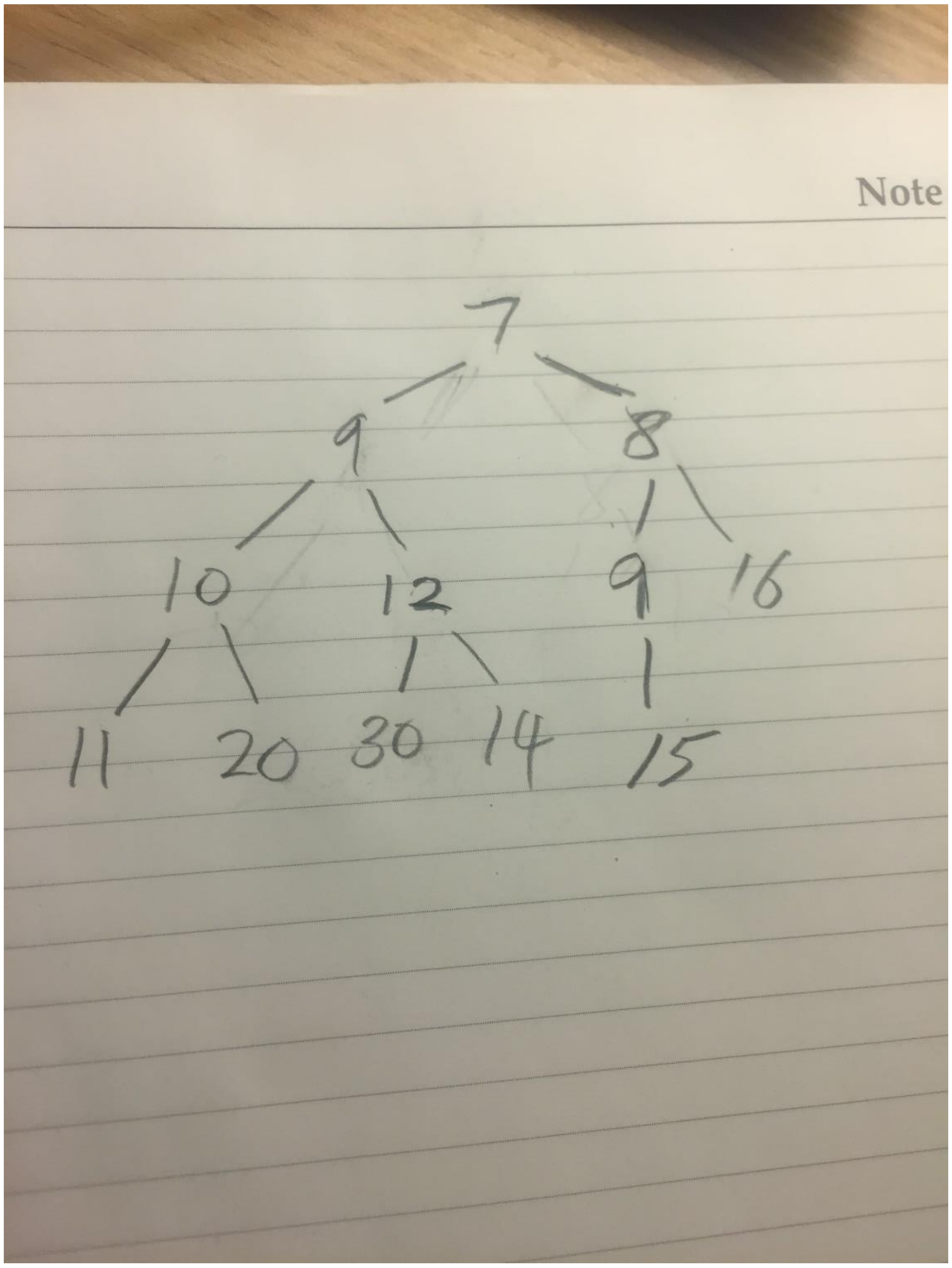
⇒ 10 9 8 5 7 2 4 3 1

⇒ Final: 10 9 8 5 7 2 4 3 1

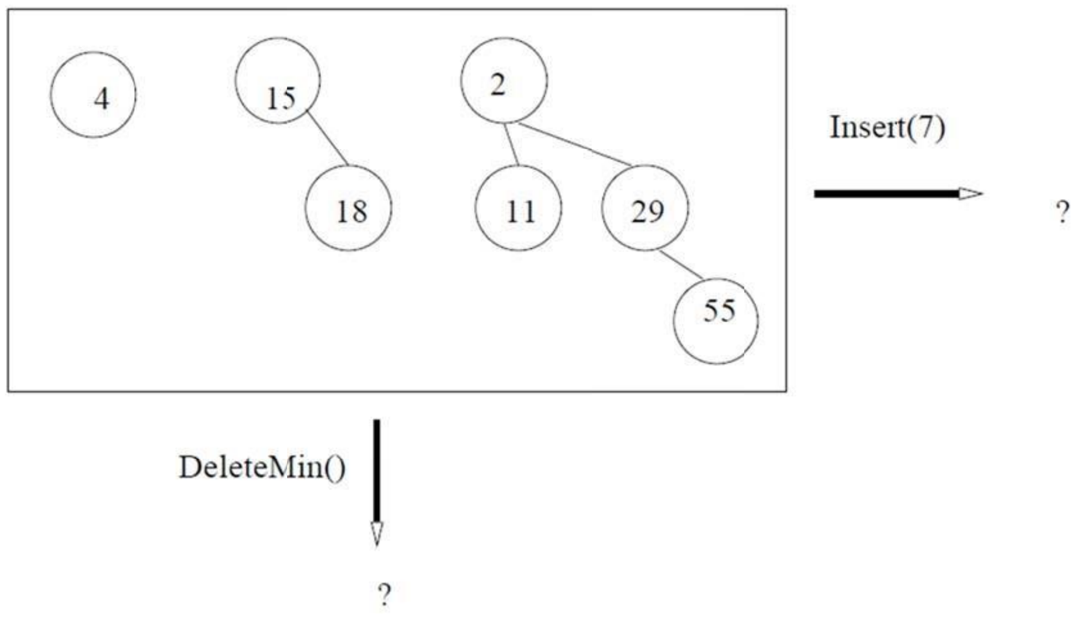
10
9
8
5
7
2
4
3
1

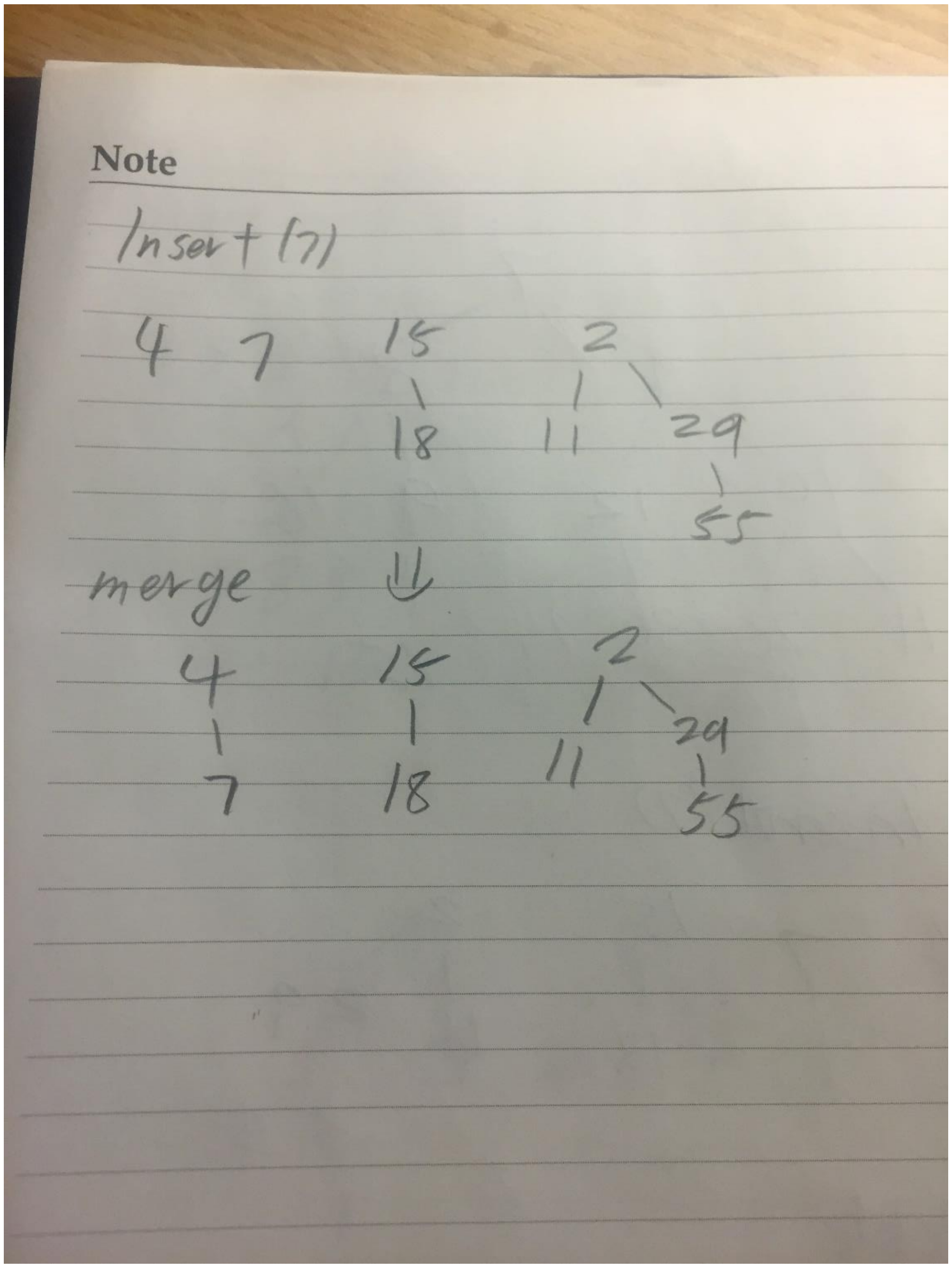
6. Show the resulting binary heap after performing DeleteMin:


DeleteMin()
?



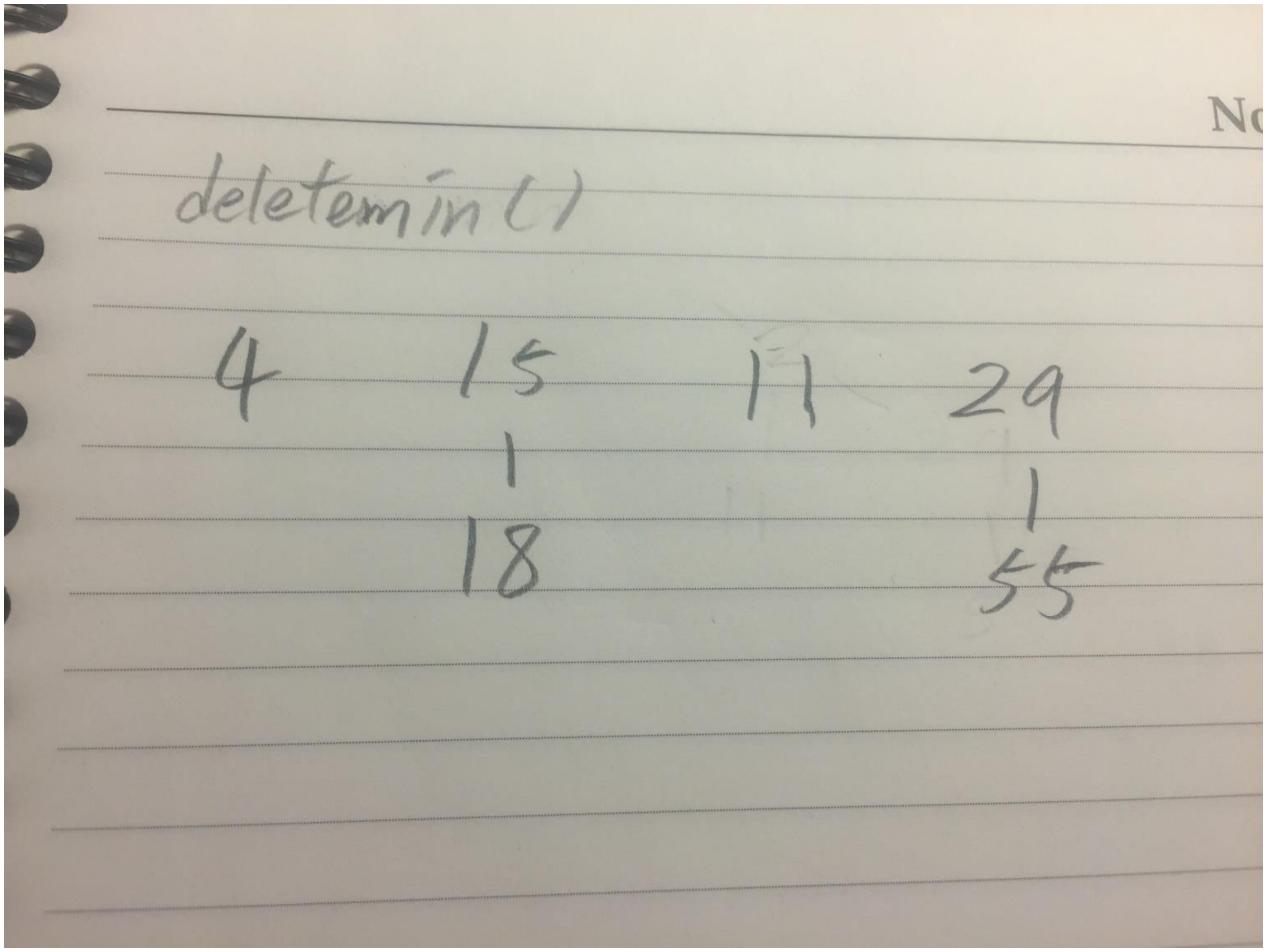
7. Perform the following two operations on the binomial heap, each independently starting with the following input binary heap:


4
15
18
2
11
29
55
Insert(7)
?
DeleteMin()
?

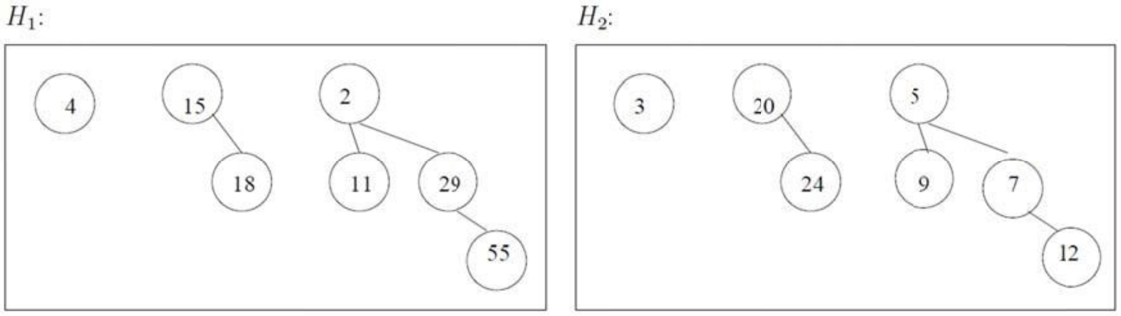


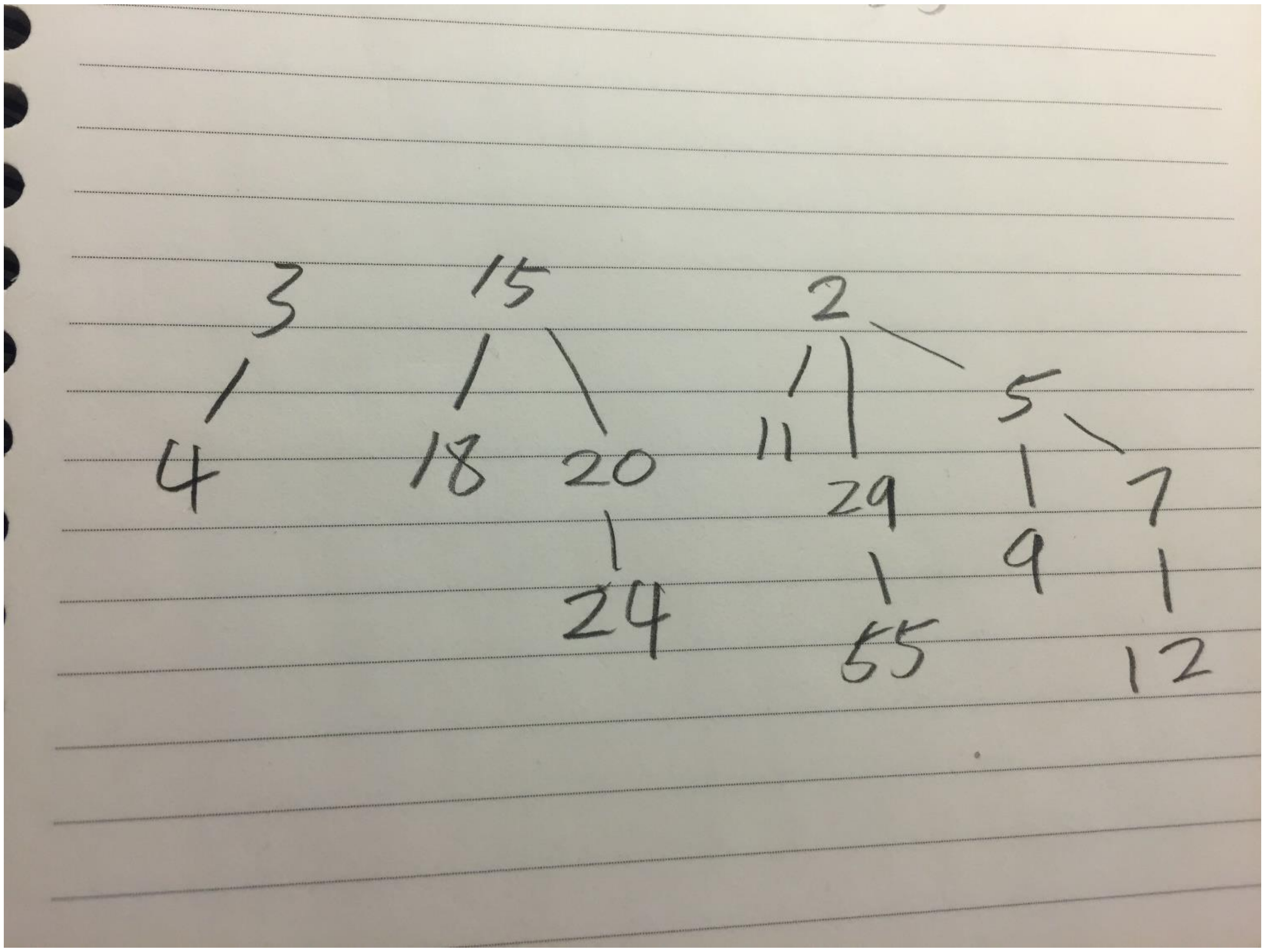
Note

Insert (7)



8. Merge the following two binomial heaps, H1 and H2:




3
4
15
18
20
24
2
11
29
55
5
9
7
12