

# Assignment 3

● Graded

## Student

IBRAHIM ALSHAYEA

## Total Points

73 / 100 pts

## Question 1

(Binary Heaps) Consider a perfect binary tree of height  $h \geq 0$ .

23 / 25 pts

- 1.1 a. (5 points) How many nodes does the tree have (in terms of  $h$ )?

5 / 5 pts

✓ - 0 pts Correct

- 5 pts Incorrect, it is  $2^{h+1} - 1$

- 1.2 b. (20 points) Show that applying the bottom-up algorithm for building a heap on a perfect binary tree costs  $\Theta(n)$  element comparisons, where  $n$  is the number of nodes in the perfect binary tree.

18 / 20 pts

Part 1: Argument leading to the summation

+ 10 pts Complete and correct explanation with clear justification.

✓ + 8 pts Mostly correct explanation with minor justification issues.

+ 4 pts Partially correct explanation with major gaps in justification.

+ 0 pts No justification or entirely incorrect explanation.

Part 2: The summation

✓ + 4 pts Correct formulation of the summation.

+ 2 pts Minor mistake in the formulation of the summation.

+ 0 pts Major mistake or completely incorrect formulation of the summation.

Part 3: Summation Evaluation

✓ + 6 pts Correct solution of the summation.

+ 4 pts Small mistake in the solution of the summation.

+ 2 pts More than one small mistake in the solution of the summation.

+ 0 pts Entirely incorrect solution of the summation.

Question 2

Sort the following array using heap sort algorithm. [77, 45, 38, 28, 12, 17, 9]

0 / 25 pts

- 0 pts Correct

✓ - 25 pts Entirely incorrect or no solution

- 20 pts Correct until [45, 28, 38, 9, 12, 17, 77] or until [9]

- 15 pts Correct until [38, 28, 17, 9, 12, 45, 77] or until [9,12]

- 10 pts Correct until [28, 12, 17, 9, 38, 45, 77] or until [9,12,17]

- 5 pts Correct until [17, 12, 9, 28, 38, 45, 77] or until [9,12,17,28]

- 3 pts Missed last step/2 steps

Question 3

Insert 12, 29, 23, 39, 43, 26, 48, 27, 20, 60, 57, 50 in a B tree of order M=3

30 / 30 pts

✓ - 0 pts Correct

- 15 pts Major error in the tree

- 20 pts Wrong insertion and number of nodes

- 5 pts minor error in the tree

- 30 pts No solution

- 2 pts one wrong node

- 15 pts No steps

- 10 pts missing steps

- 30 pts Cheating

Excellent solution, good luck in your final exams.

Question 4

B Trees

20 / 20 pts

4.1 a. (10 points) Delete the key 91 in the following B-tree of order 5

10 / 10 pts

✓ - 0 pts Correct

- 2 pts Minor mistake

- 5 pts Major mistake

- 10 pts Totally wrong/No solution

- 10 pts Cheating

4.2 b. (10 points) Delete the key 40 in the following B-tree of order 3

10 / 10 pts

✓ - 0 pts Correct

- 2 pts Minor Mistake

- 5 pts Major Mistake

- 10 pts Totally Wrong/No solution

- 10 pts Cheating

Question assigned to the following page: [1.1](#)

## Question 1.

a.

We know that at each level, the number of nodes is  $2^K$ , where  $K$  is the level number. Also, we know that the height  $h$  is the ~~path~~<sup>longest path</sup> from the root node to a leaf node, that is  $K+1$

So ...

$$\text{Total nodes} = \sum_{K=0}^h 2^K = \frac{2^{h+1} - 1}{2 - 1} = \underline{\underline{2^{h+1} - 1}}$$

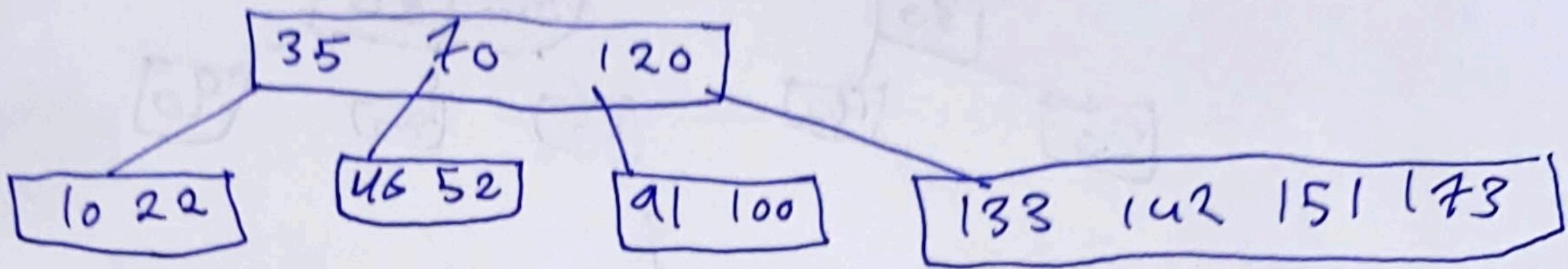
Question assigned to the following page: [4.1](#)

Question IV.      maxflow condition  $\lceil m/2 \rceil - 2$  keys  
 min-keys:  $\lfloor m/2 \rfloor - 1$

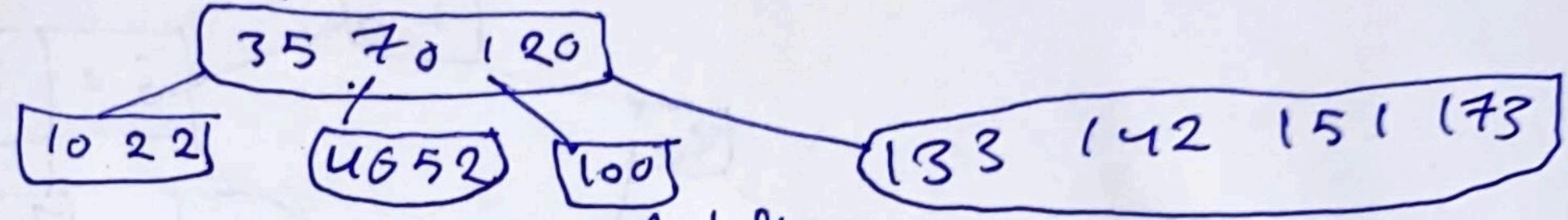
a. Delete the key 91, offset 5

maxflow:  $5/2 - 2 = 1$

min-keys:  $5/2 - 1 = 2$



Delete 91



underflow

left rotation

~~left rotation~~



1622

4652

100 120

142 151 173

Question assigned to the following page: [1.2](#)

## Question I

b. the algorithm: private void buildHeapBottomUp()  
for int i = heapSize/2; i >= 1; i-- {  
    percolateDown(i);}

}

h: height of the heap  
k: the level index

At each level of the heap, there are  $2^k$  nodes  
The number of levels in the heap is  $h = \log_2 n$

total number of comparisons:  $\sum_{k=0}^h 2^k$

$$= \frac{2^{h+1} - 1}{2 - 1} = 2^{h+1} - 1$$

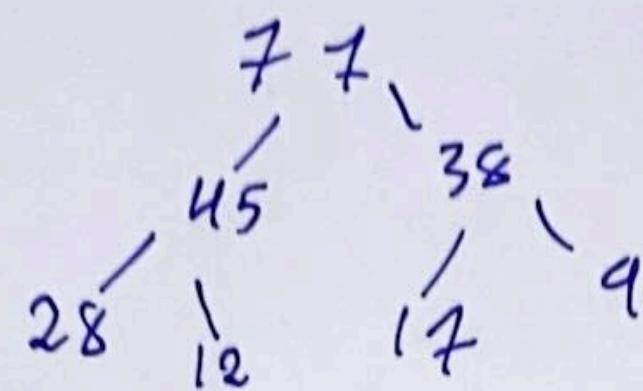
We know that  $h = \log_2 n$ , so

$$\text{Total number of comparisons} = 2^{\log_2 n + 1} - 1 = 2n - 1$$

Therefore we can say that the number of comp  
is.  ~~$\Theta(2)$~~   $\Theta(n)$

## Question II:

1. Build a max heap from the array  $[77, 45, 38, 28, 12, 17, 9]$  that is;



2. Swap the root with the last element after swapping:

$[9, 45, 38, 28, 12, 17, 77]$

3. Reduce the size of the heap, the last elements (77) is considered sorted, so the heap size is now 6

4. Heapsify the root:  $[45, 28, 38, 9, 12, 17, 77]$

5. Repeat these steps until the heap size is 1

After each iteration: swap

- Swap root with last element
- Reduce heap size
- Heapsify Root

[17, 28, 38, 9, 12,

12, 9, 38, 17, 28, 45, 77]

9, 12, 38, 17, 28, 45, 77]

12, 17, 9, 38, 17, 28, 45, 77]

17, 28, 9, 12, 38, 45, 77]

12, 9, 17, 28, 38, 45, 77]

9, 12, 17, 28, 38, 45, 77]

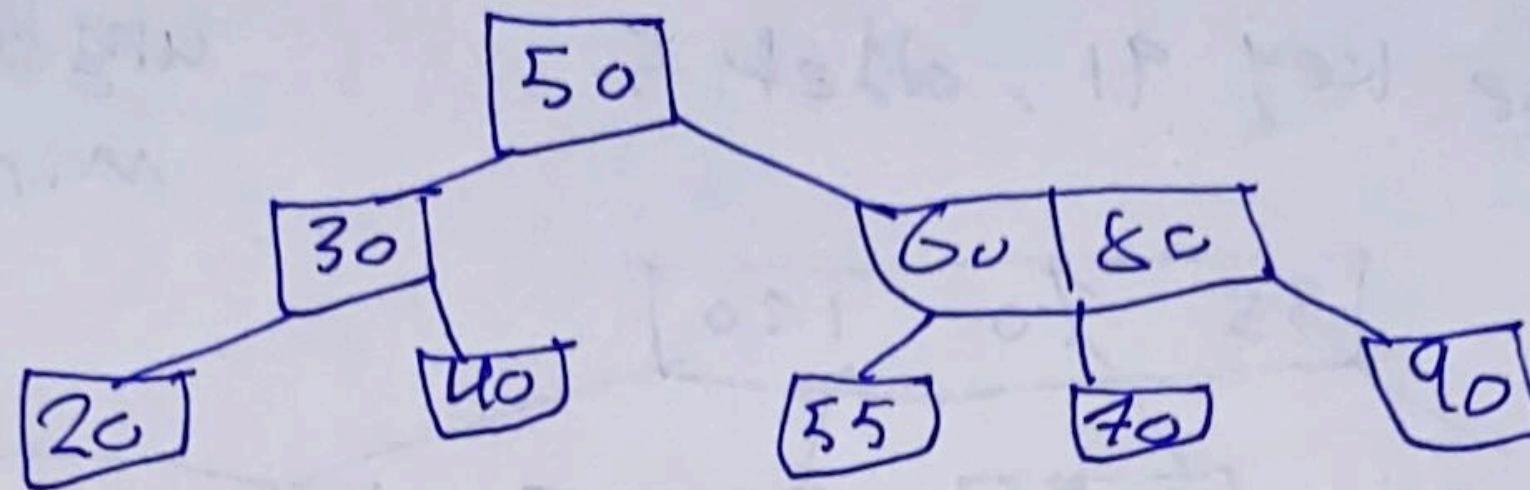
So, the sorted array is [9, 12, 17, 28, 38, 45, 77]

[9, 12, 17, 28, 38, 45, 77]

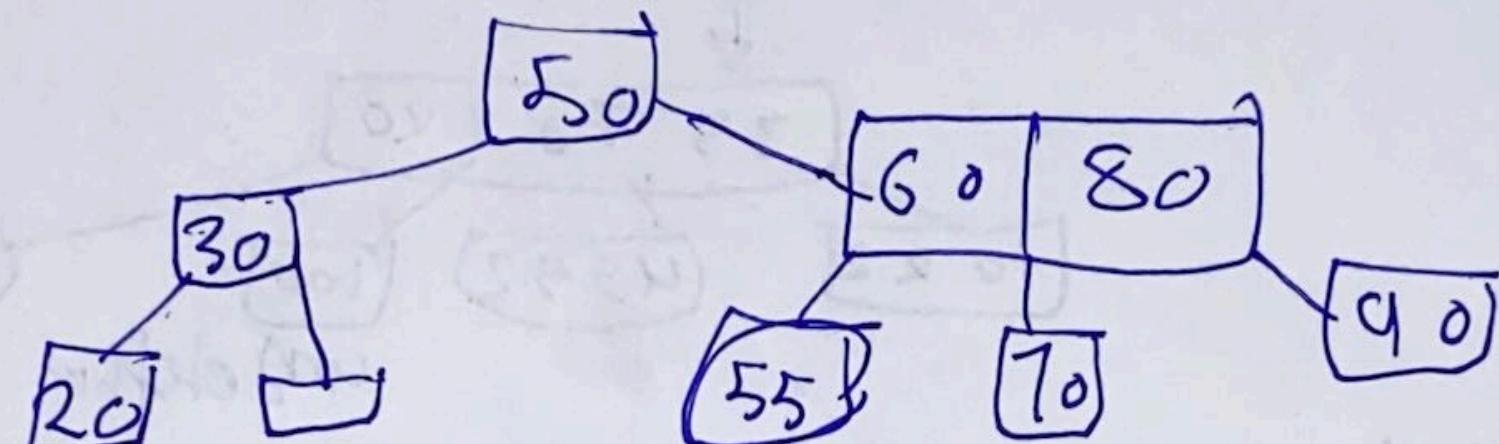
Question assigned to the following page: [4.2](#)

b. Delete no , offset 3

overflow cond :  $3/2 - 2 = 0$   
Min-Kees :  $3/2 - 1 = 1$

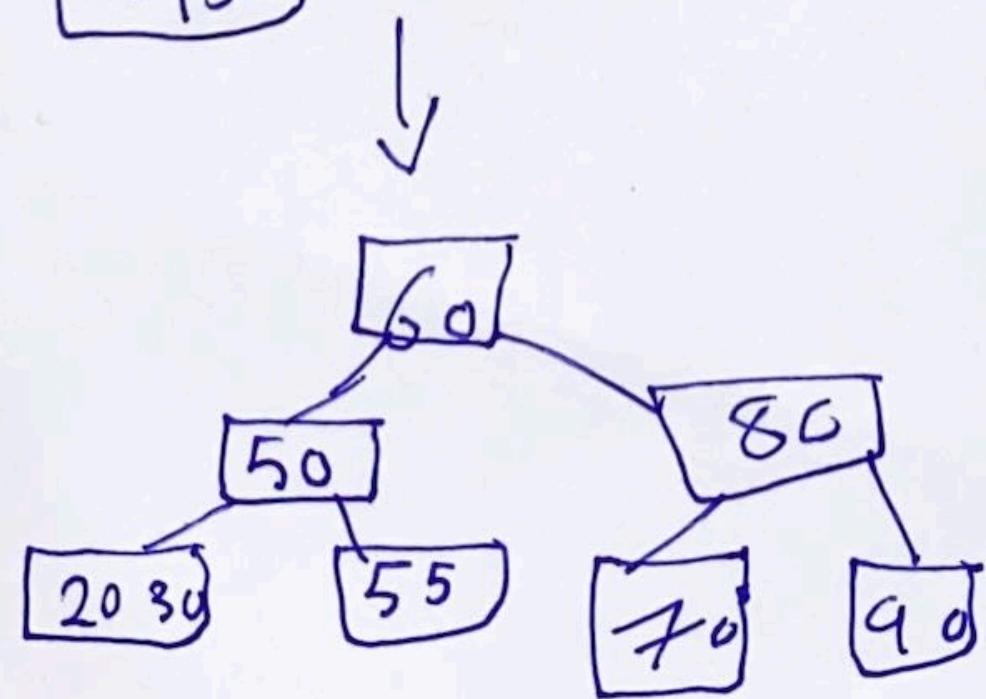
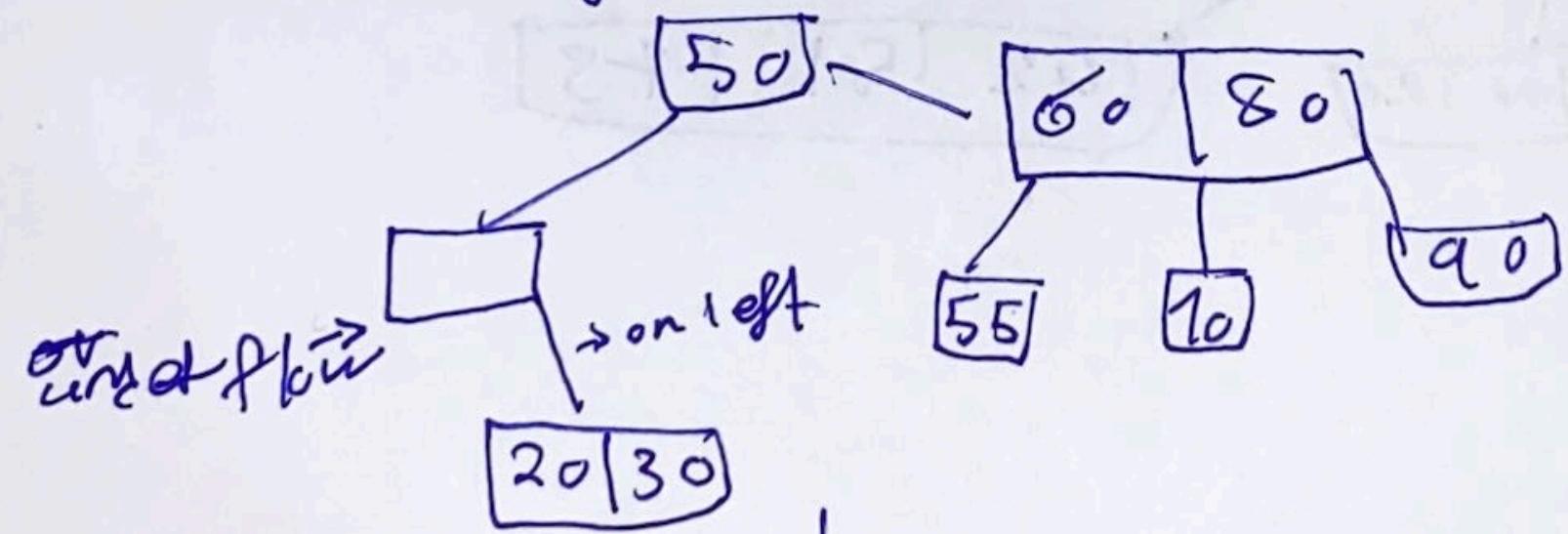


Delete no

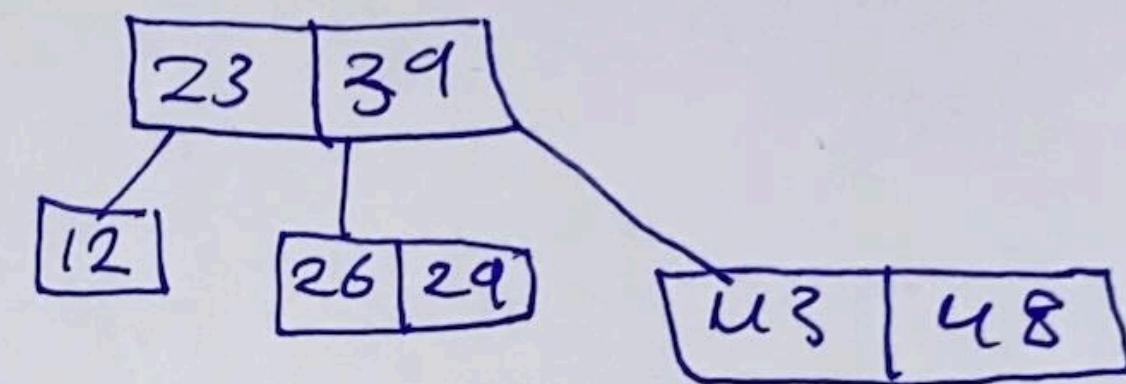
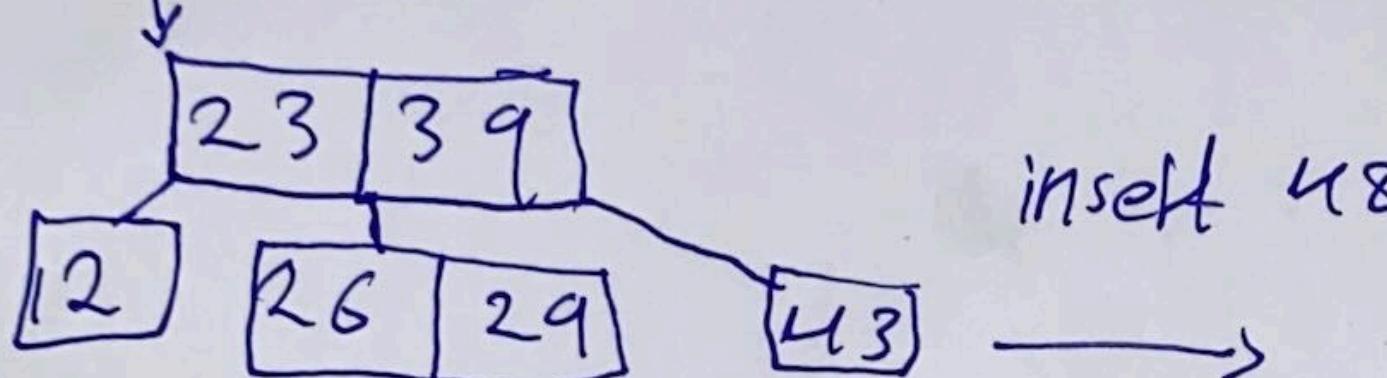
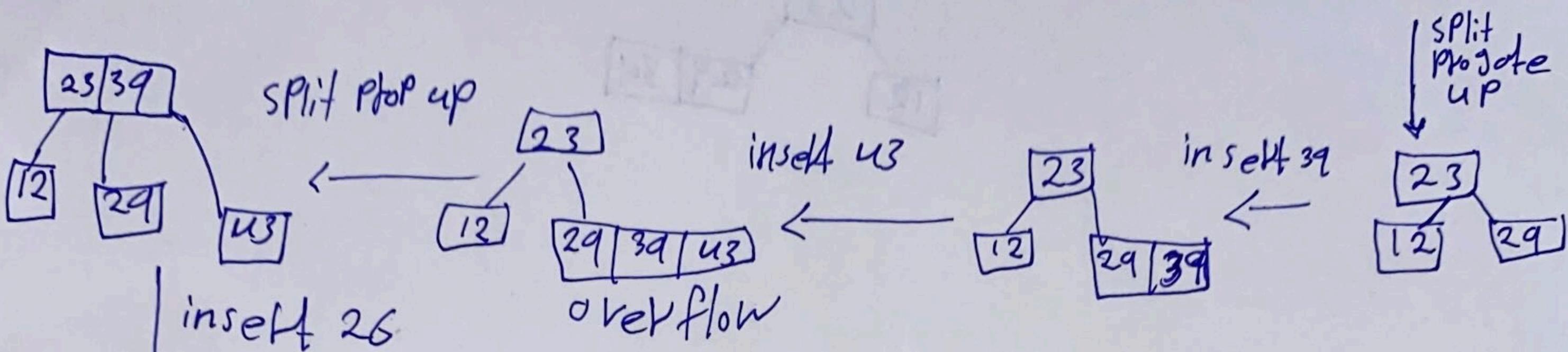
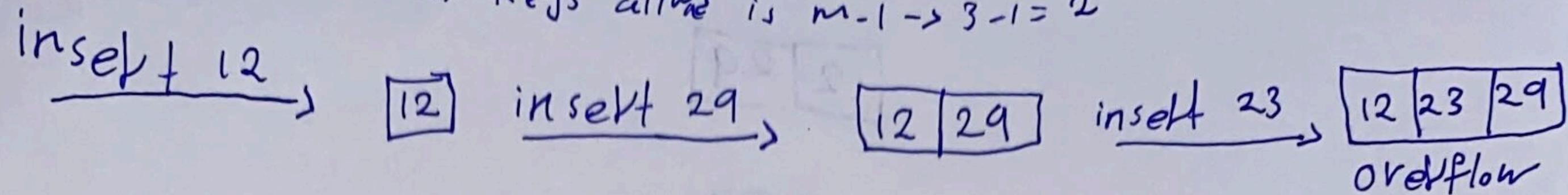


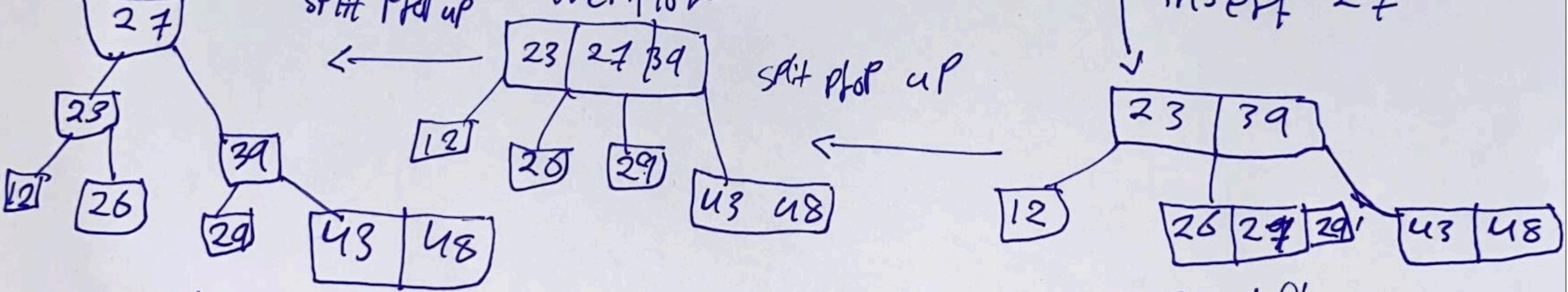
underflow condition

move 30 over 20

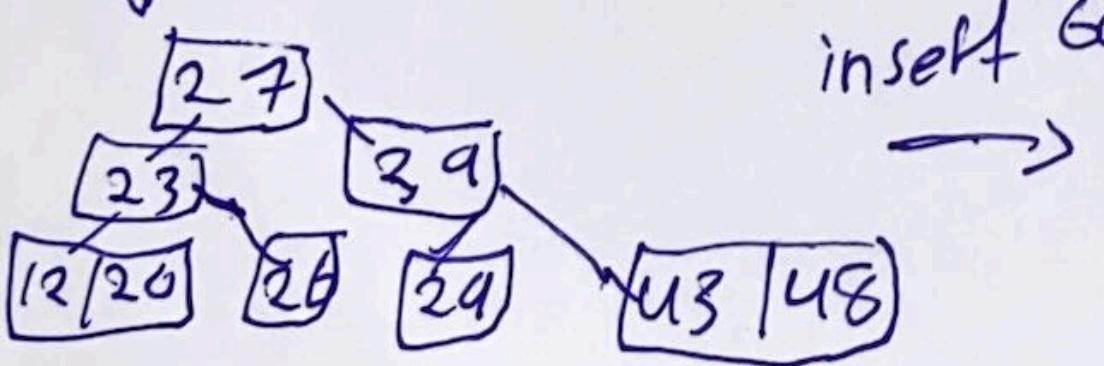


Question III: Insert 12, 29, 23, 39, 43, 26, 48, 27, 20, 60, 57, 50  
 in a B tree of order  $M=3$   
 max keys allowed is  $m-1 \rightarrow 3-1 = 2$

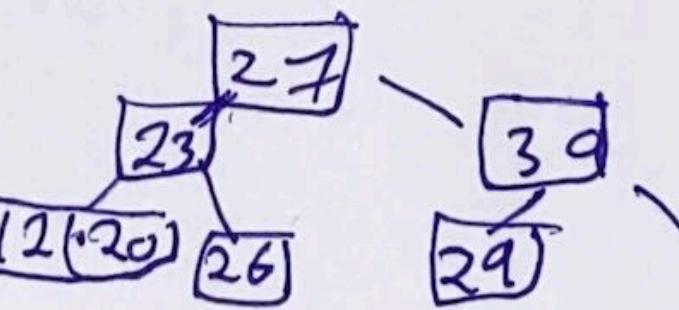




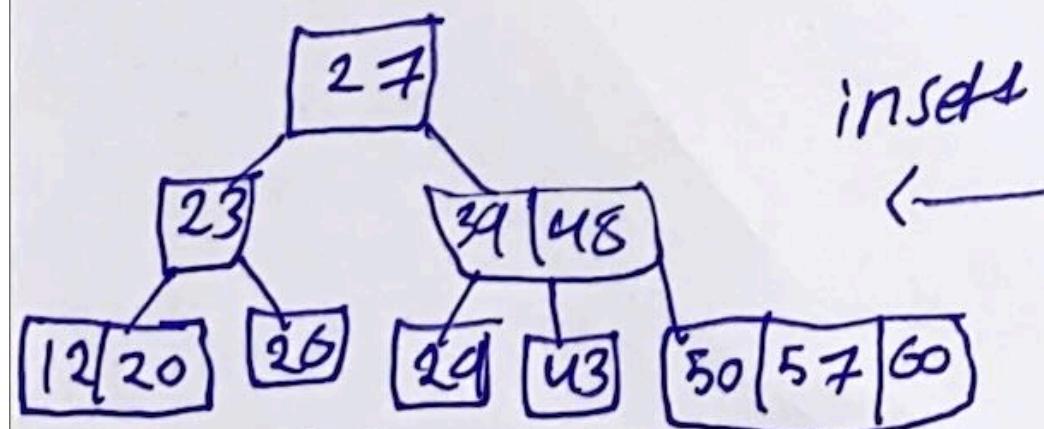
insert 20



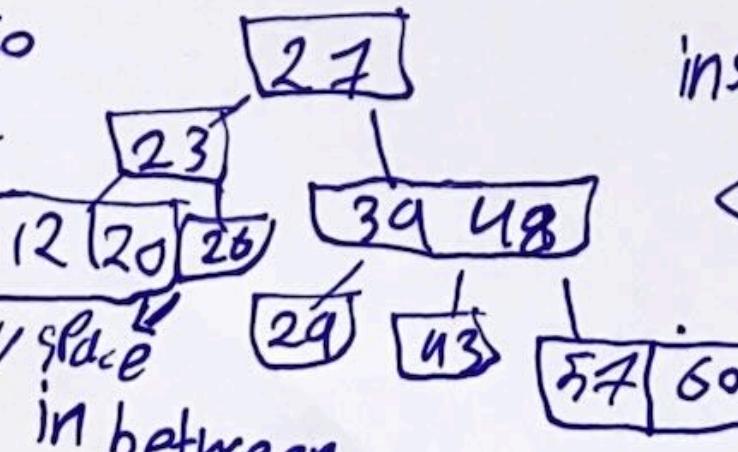
insert 60



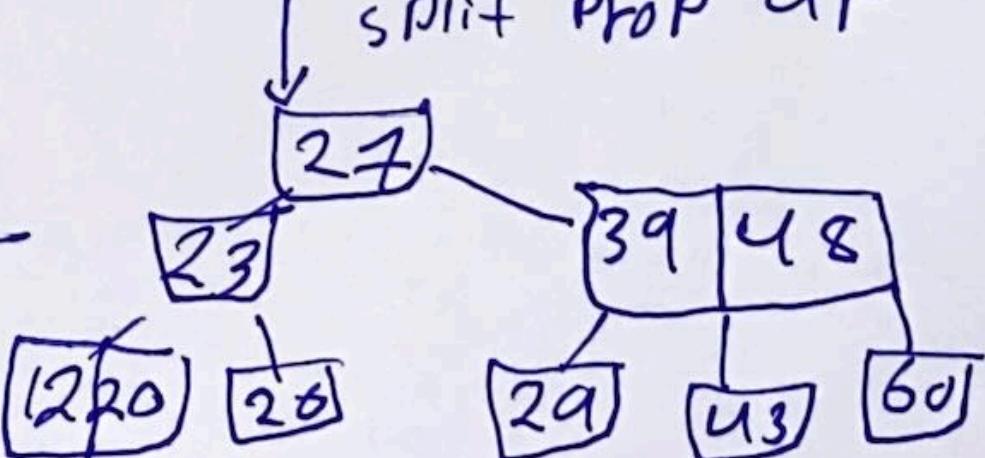
overflow



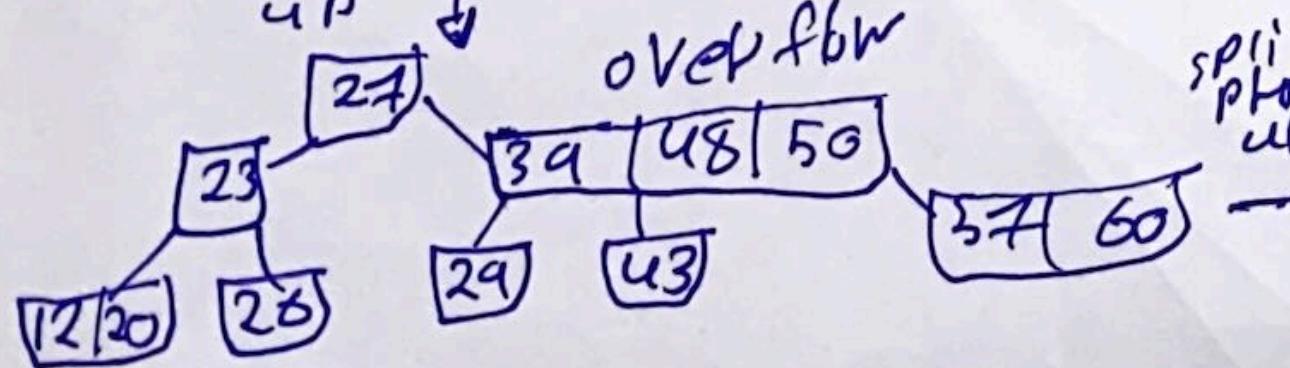
insert 50



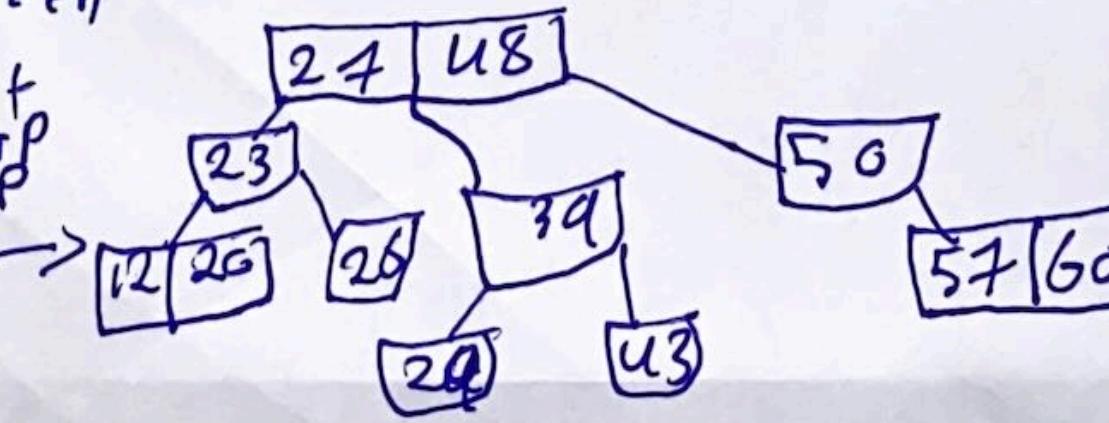
insert 57



split prop up



split prop up



→ final B tree

