Priority Queues

priority satellite data

- Each entry (key, Value)

  * Insert (key, value)

  * RemoveMin ( )

    removes and returns the entry with minimum key

  * min( )

    returns the entry with minimum key

  * size( ), isEmpty( )

Commands	Return Value
insert (5, A)	
insert (9, C)	
insert (3, B)	
min( )	(3, B)
Remove Min( )	(3, B)
insert (7, D)	
removeMin ( )	(5, A)

Heaps

- Binary Tree:

node
- key
- value
- parent
- left child
- right child

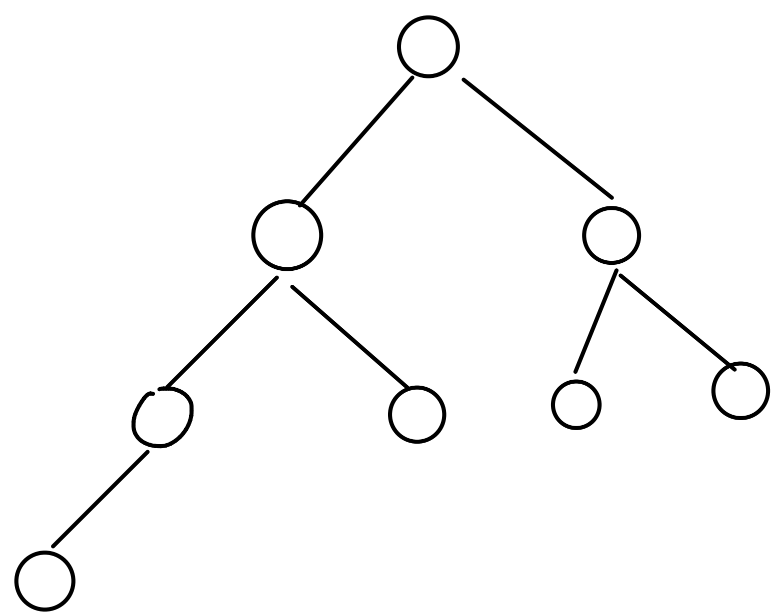
Property I

(Compare it with property of BST)

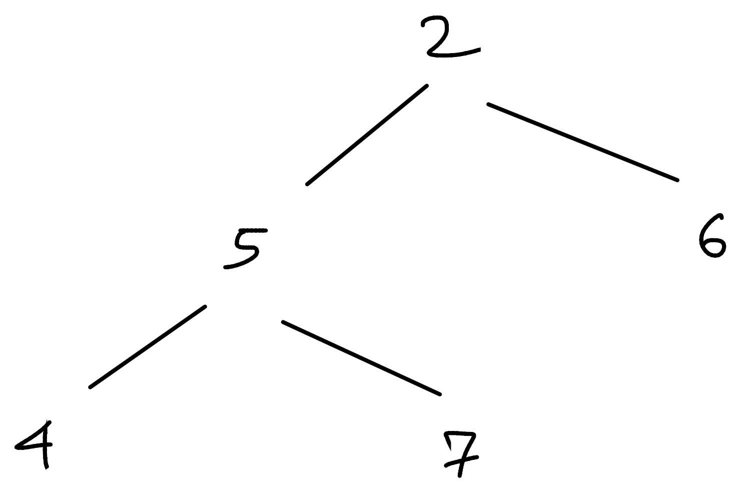
- key (parent) ≤ key (children)

Property II

- Complete Binary Tree : the last level is full from left

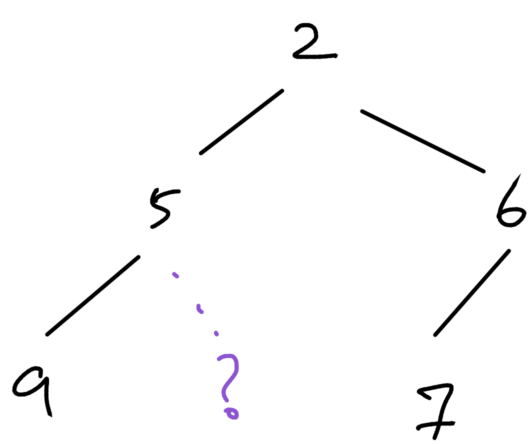


Non Example



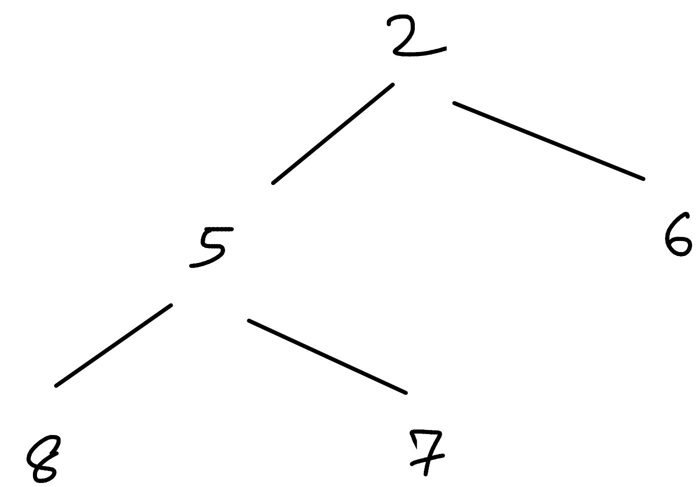
violates

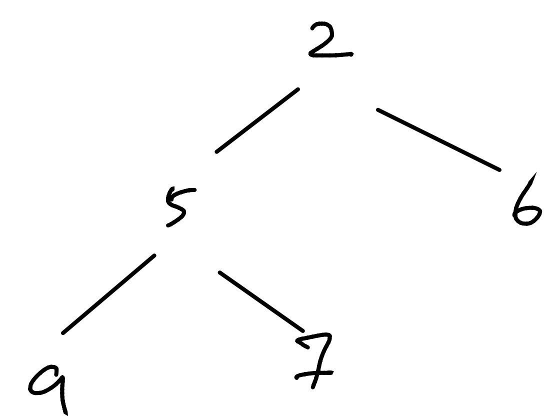
key (parent) ≤ key (children)



last level is not full from left

Example





- min() :

  Root has minimum element, can be found in $\underbrace{O(1)}_{\text{1 operation}}$

- Height is $h \leq \log n$

  Given height is h

  $n \geqslant$ minimum number of nodes in heap of ht 'h'

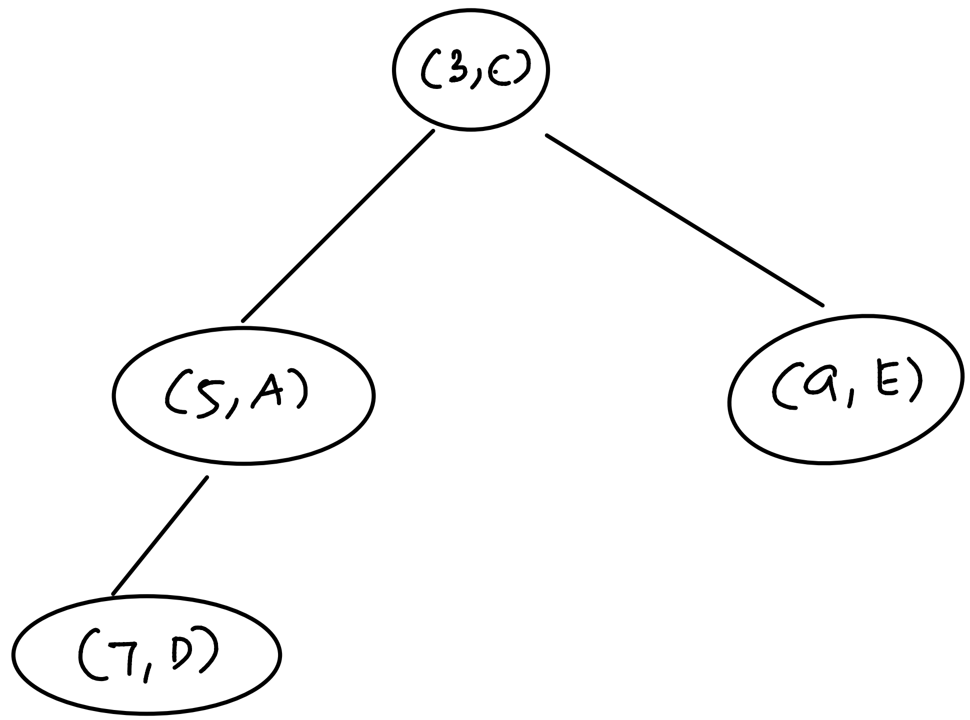
  $= 1 + 2 + \dots + 2^{h-1} + 1$

  $= 2^h - 1 + 1$

  $= 2^h$

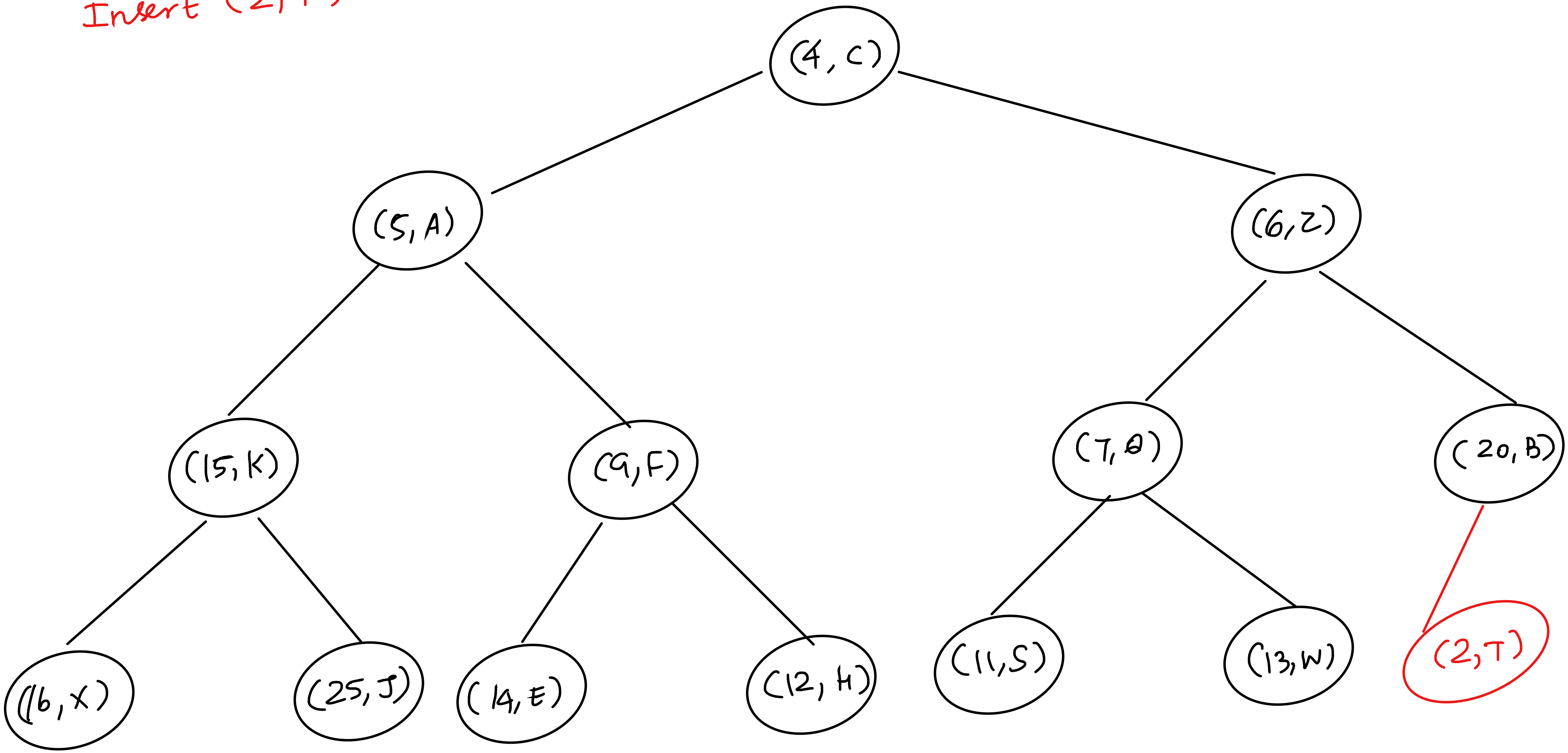
  $n \geqslant 2^h$

  $h \leqslant \log_2 n$


(3, C)
(5, A)
(9, E)
(7, D)

Insert into a Heap

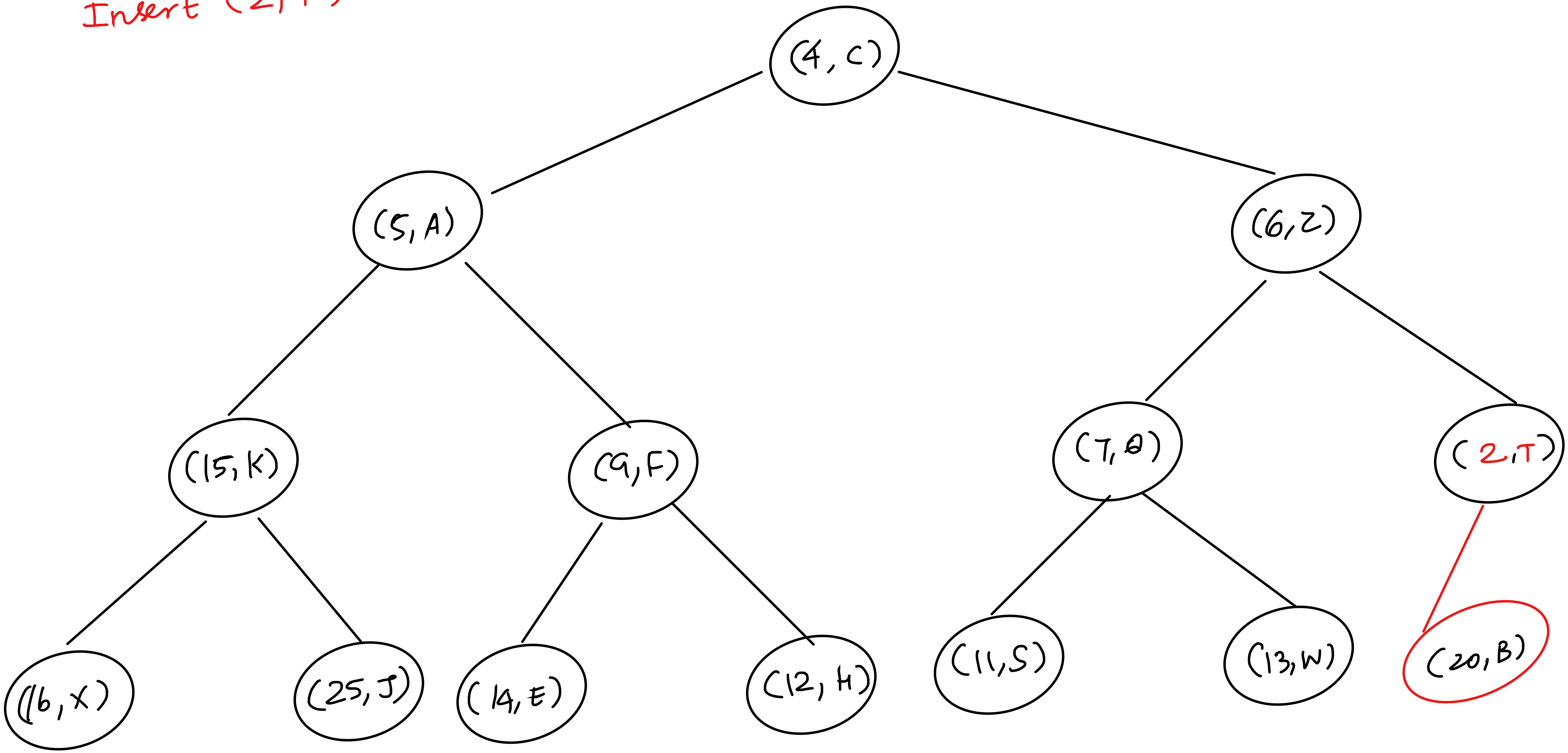
Insert (2, T)



upheap:
- Swap with parent until heap property is not violated.
- Continue till root if required

Insert into a Heap

Insert (2, T)

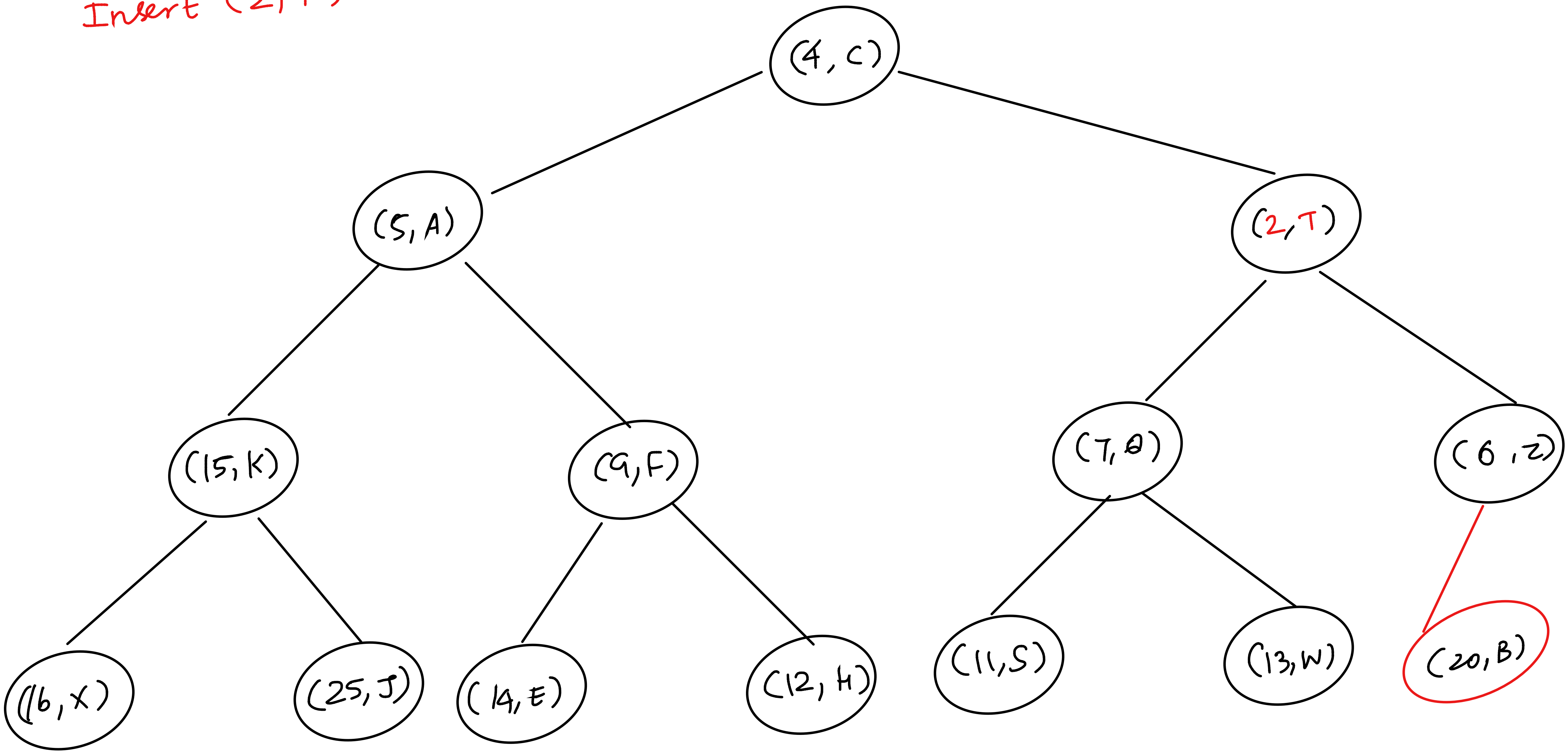


upheap: • Swap with parent until heap property is not violated.

• continue till root if required

Insert into a Heap

Insert (2, T)

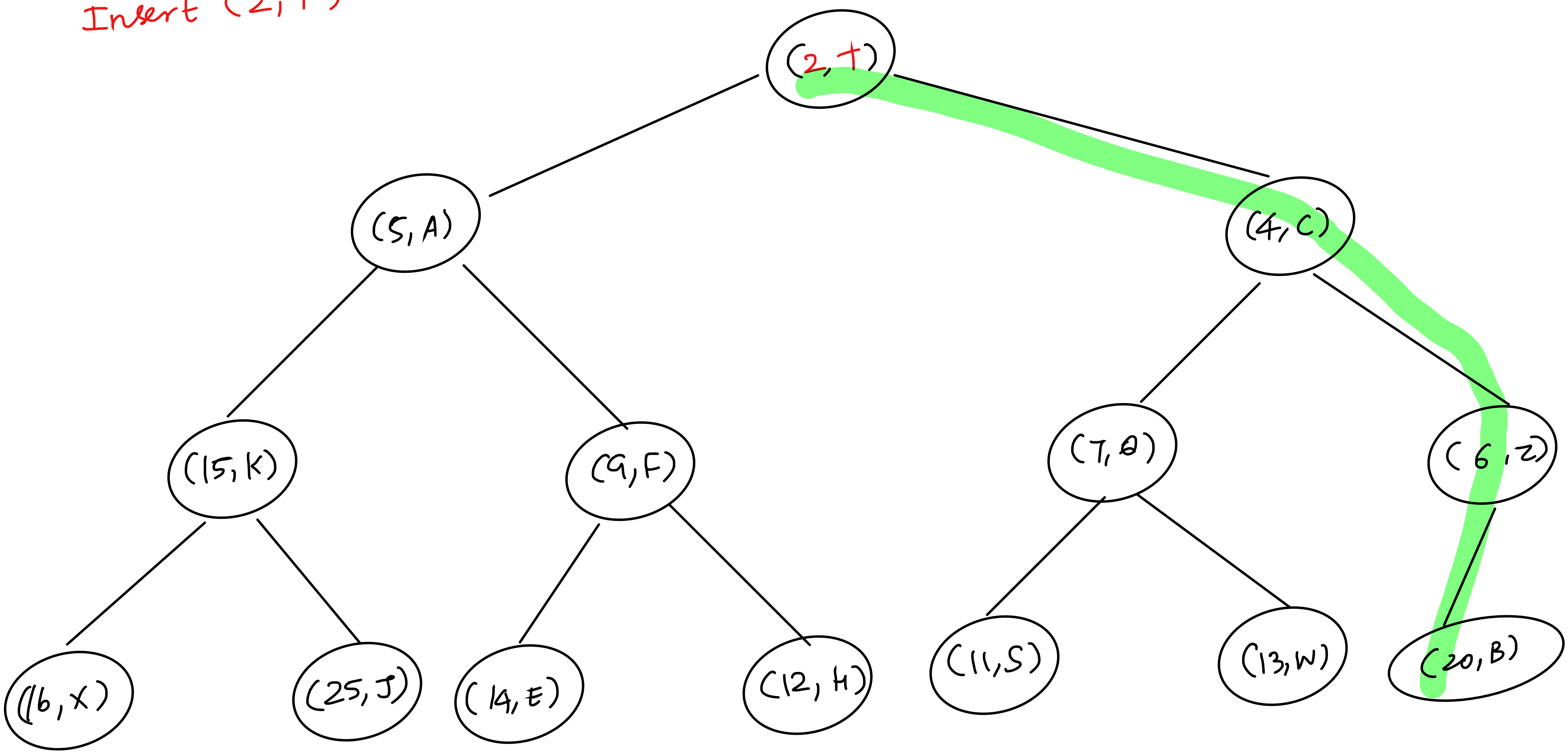


upheap: • Swap with parent until heap property is not violated.

• continue till root if required

Insert into a Heap

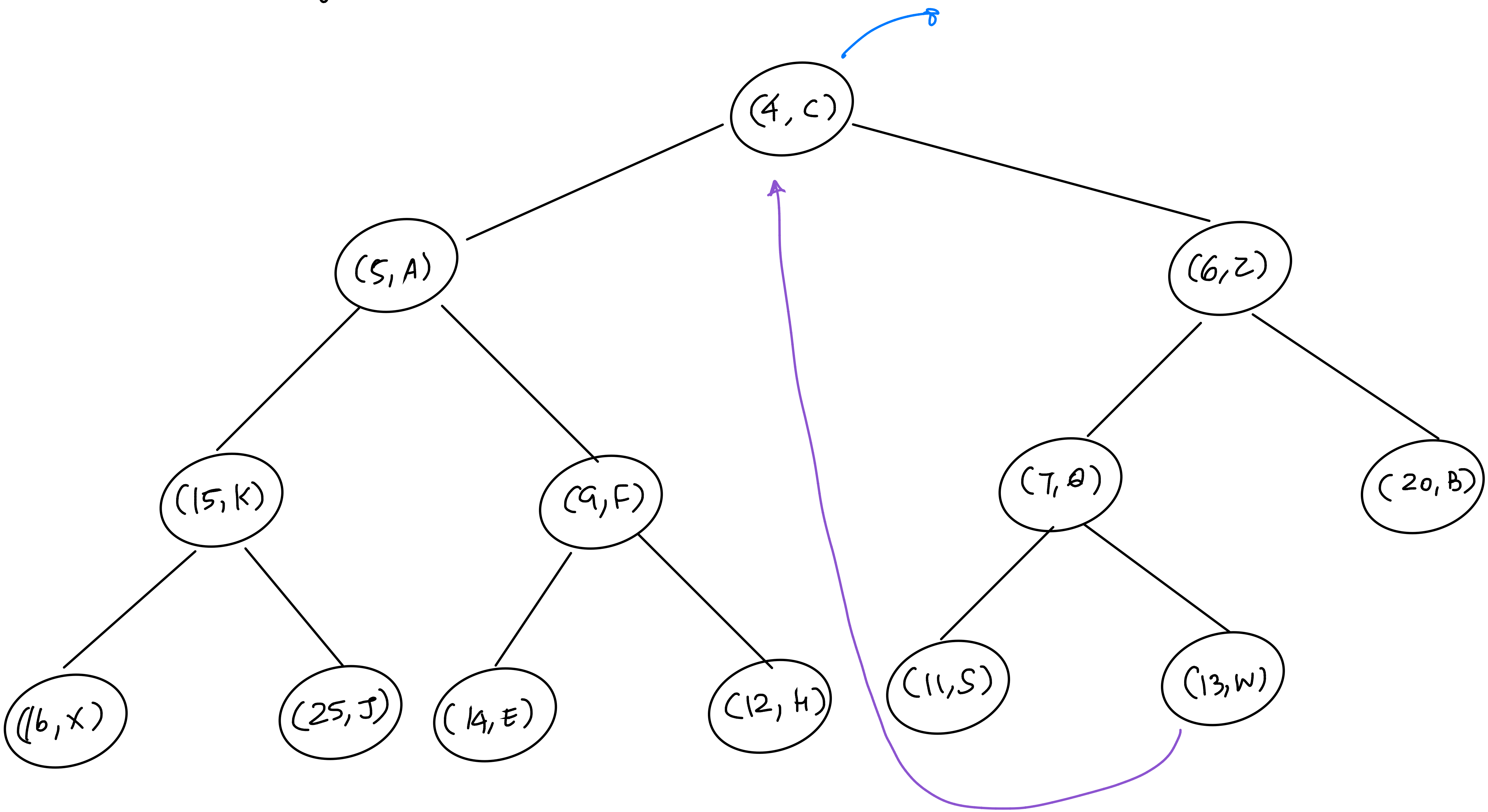
Insert (2, T)



upheap: • Swap with parent until heap property is not violated.

• continue till root if required

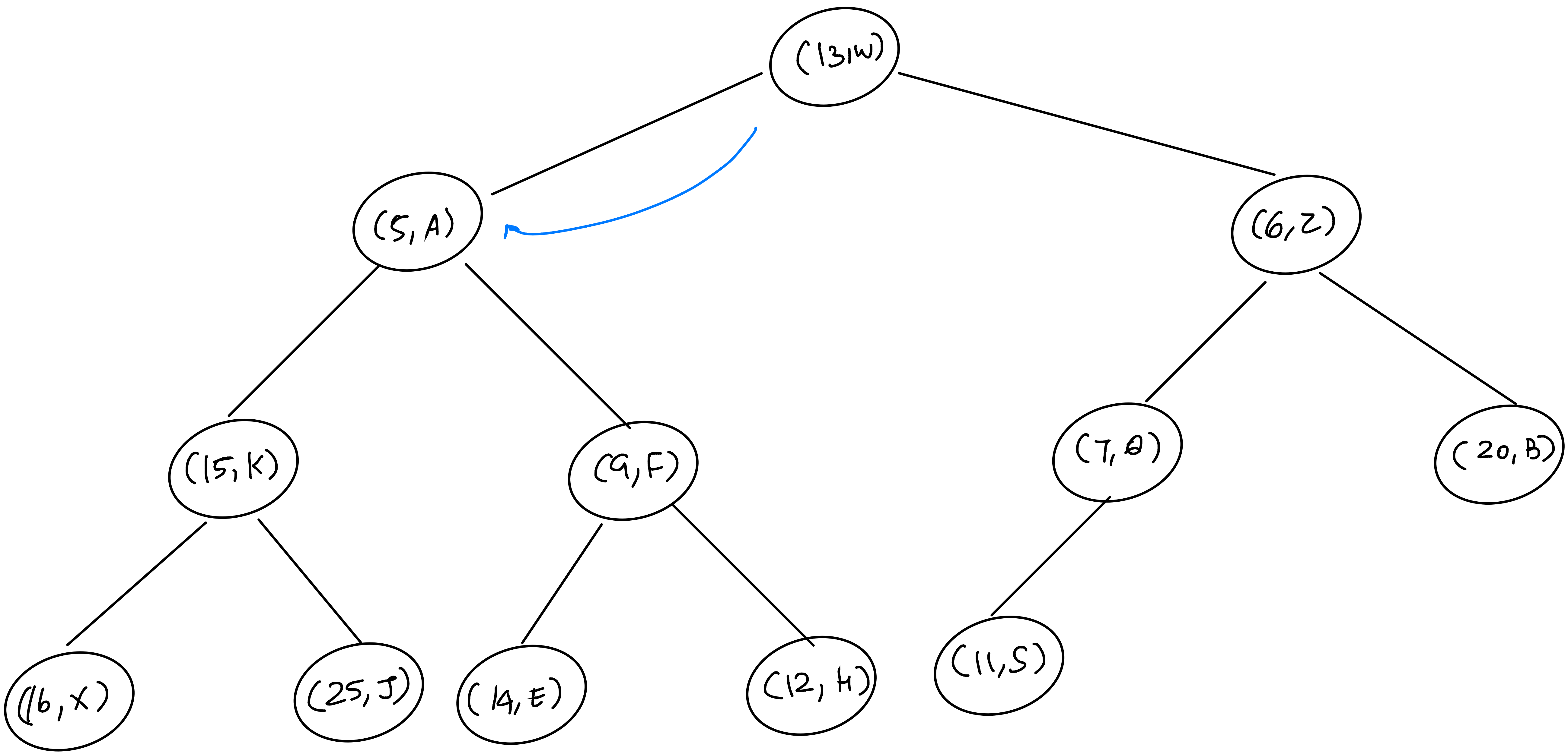
Remove Min( ) from Heap



- Remove the root
- replace the root with the right most element in the last level.

Remove Min( ) from Heap

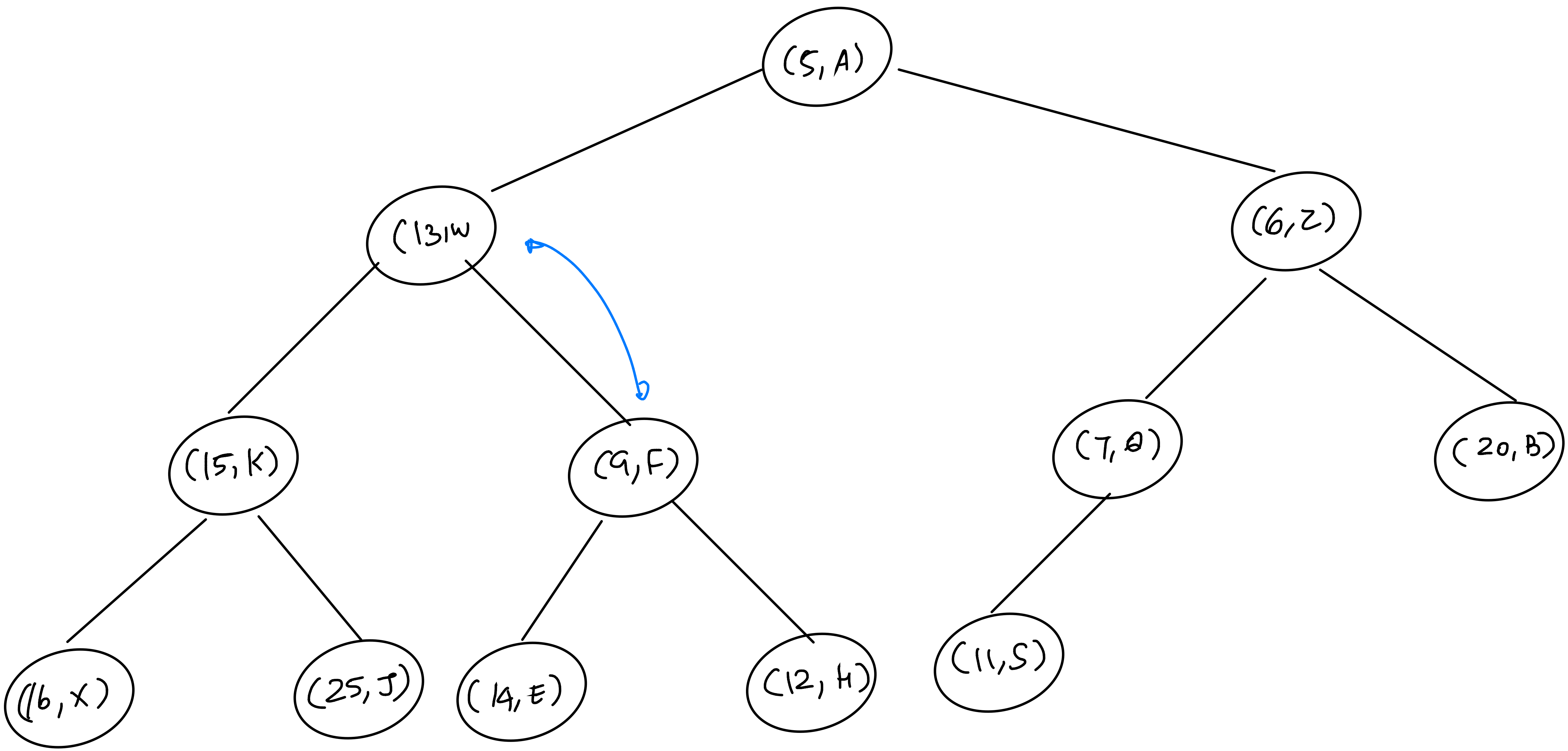
Heap property is getting violated



Perform Down heap to restore heap property

Remove Min ( ) from Heap

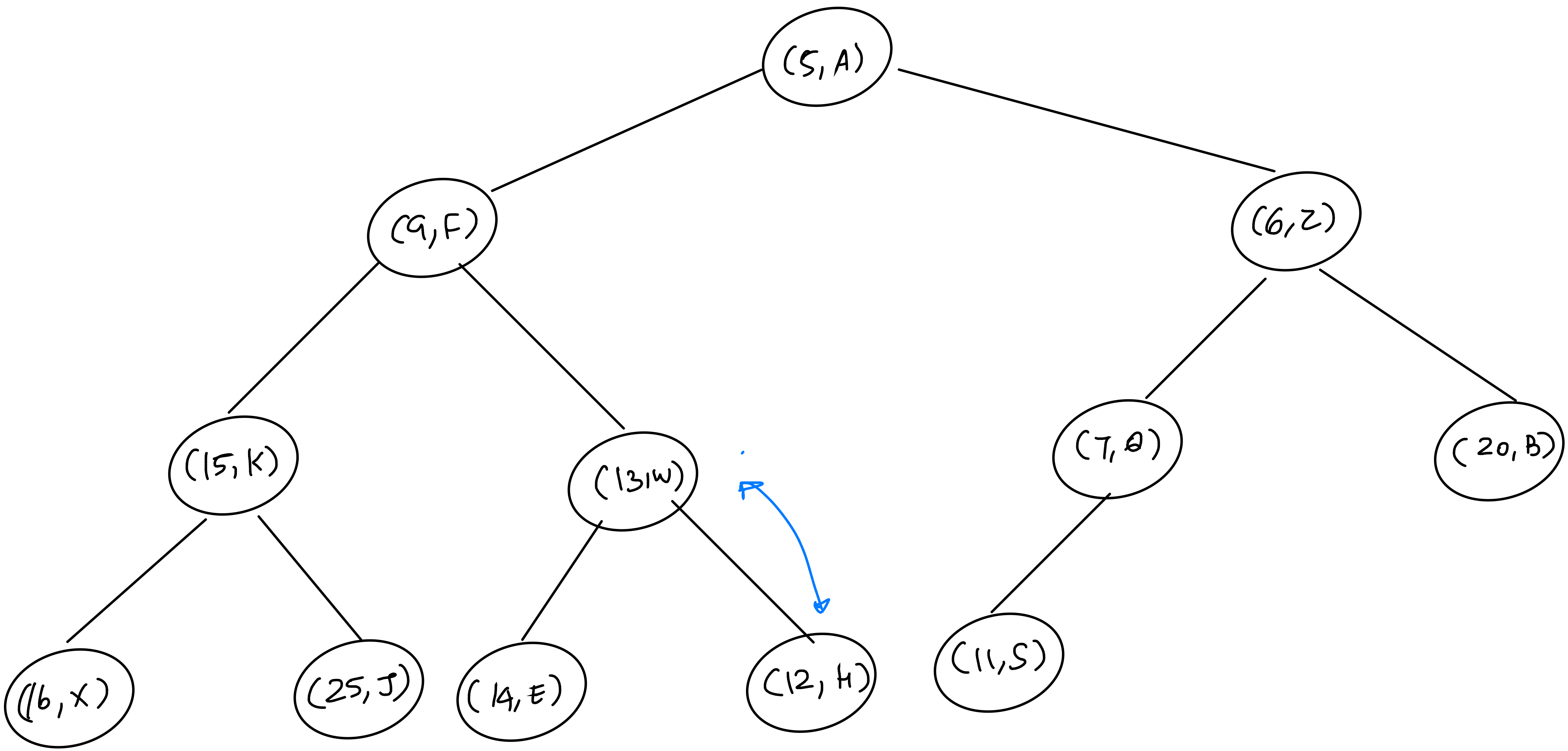
Heap property is getting violated



Perform Down heap to restore heap property

Remove Min( ) from Heap

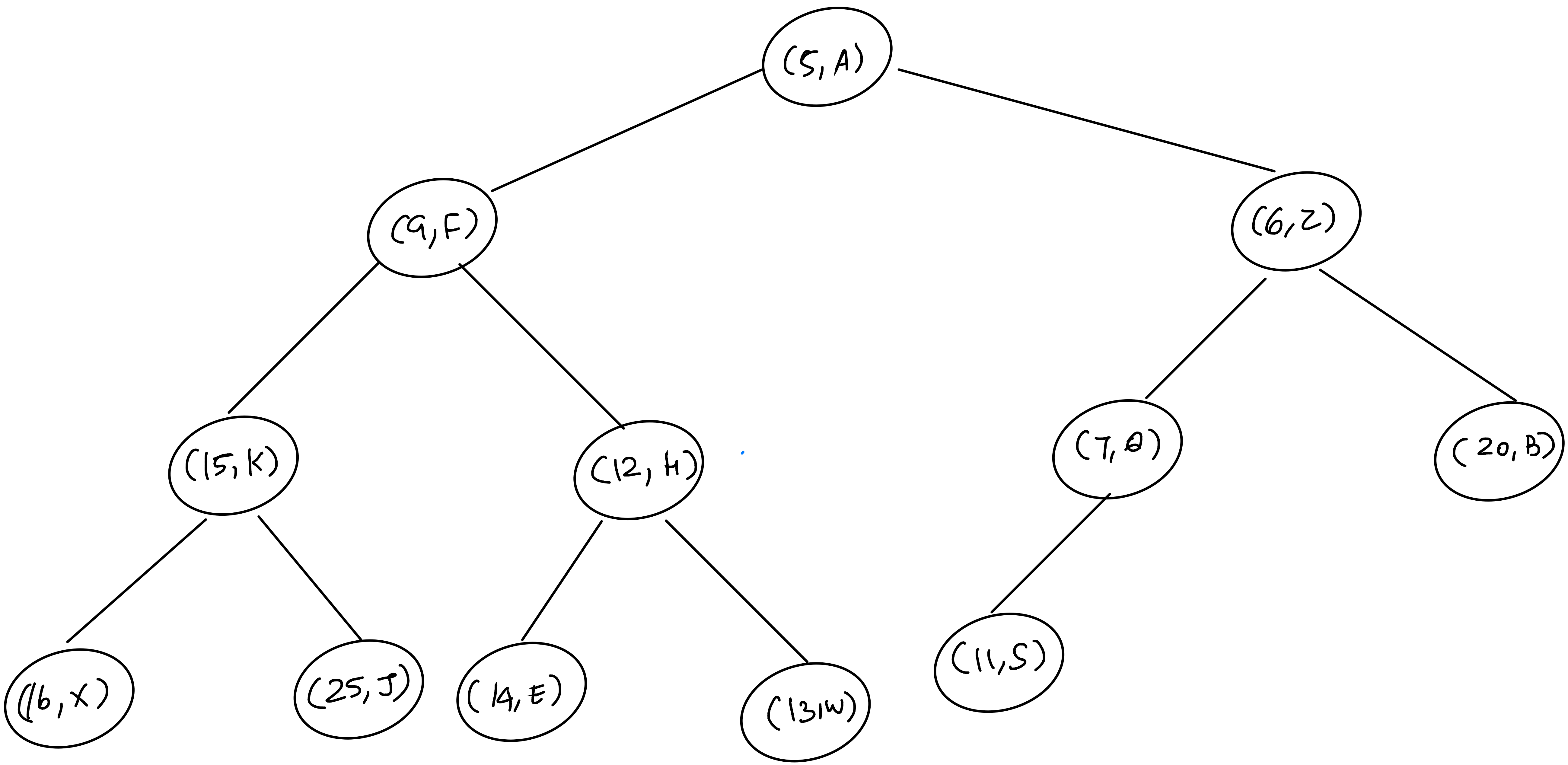
Heap property is getting violated



Perform Down heap to restore heap property

Remove Min ( ) from Heap

Heap property is getting violated



Per form Down heap to restore heap property

Qn: Given n elements, How to build heap from scratch

Ans: Add one element at a time, calling upheap

$$\log 1 + \log 2 + \cdots + \log n$$

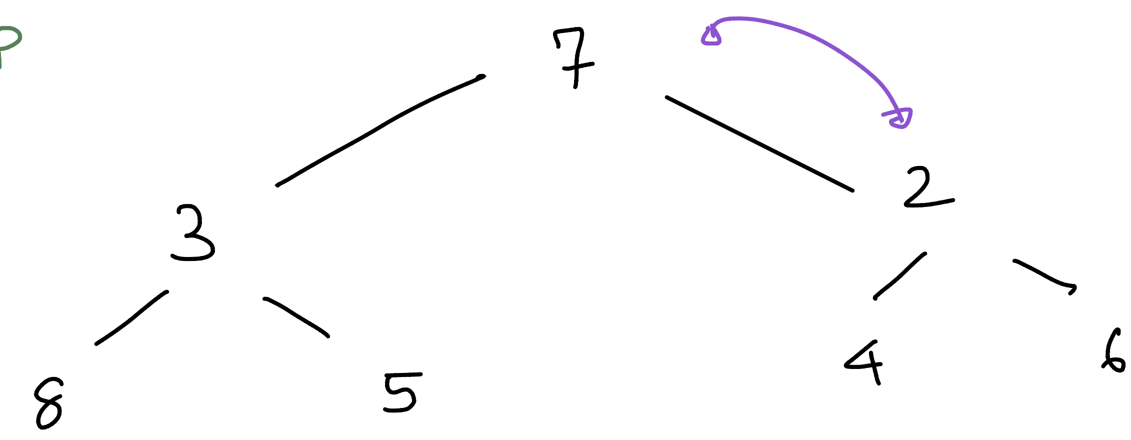
$$= \log 1 \, 2 \cdots n = \log n!$$

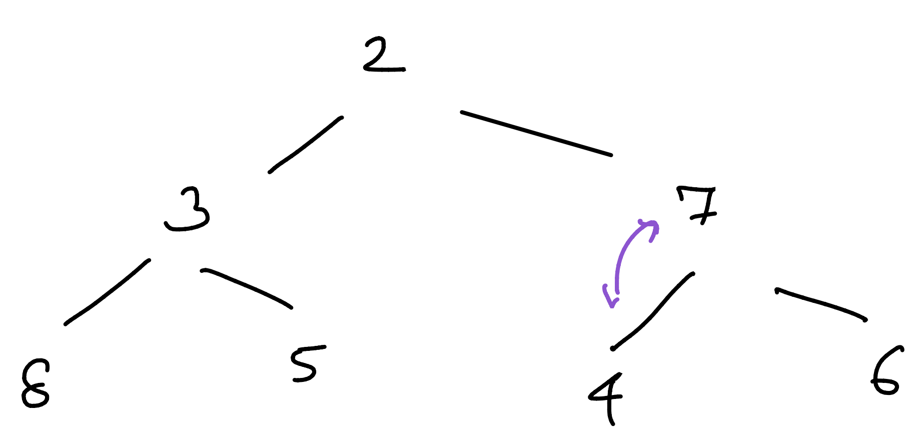
$$n! \sim n^n$$

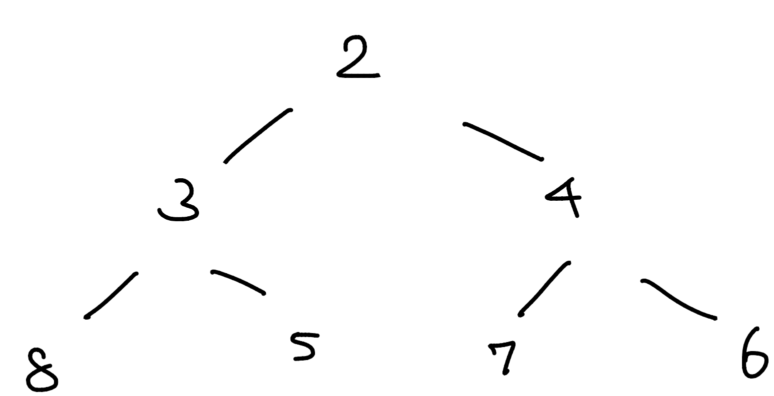
$$\log n^n = n \log n$$

Merging Two Heaps with new element at root

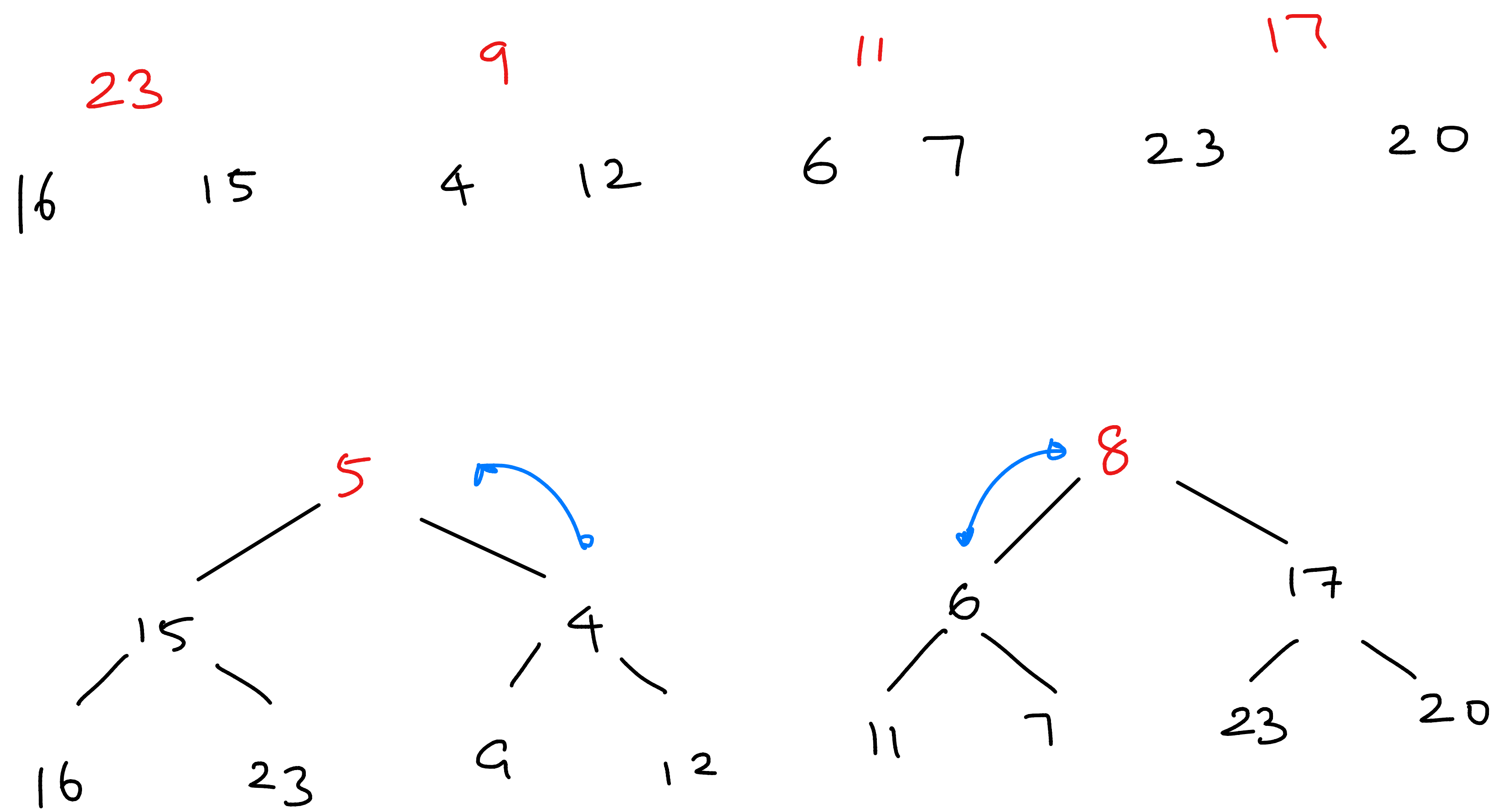
Do down heap

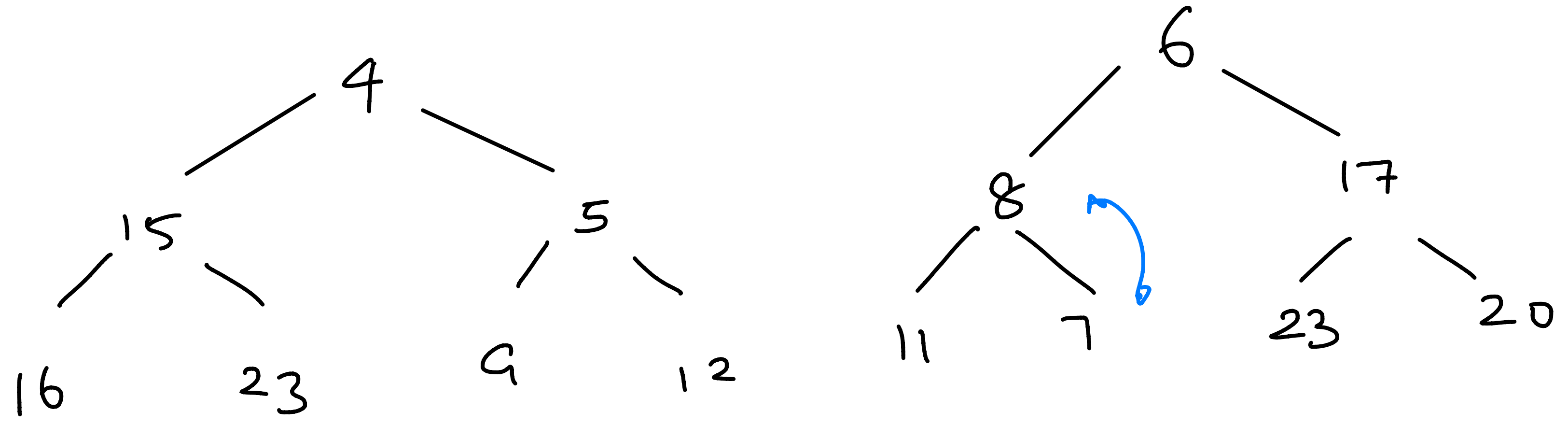




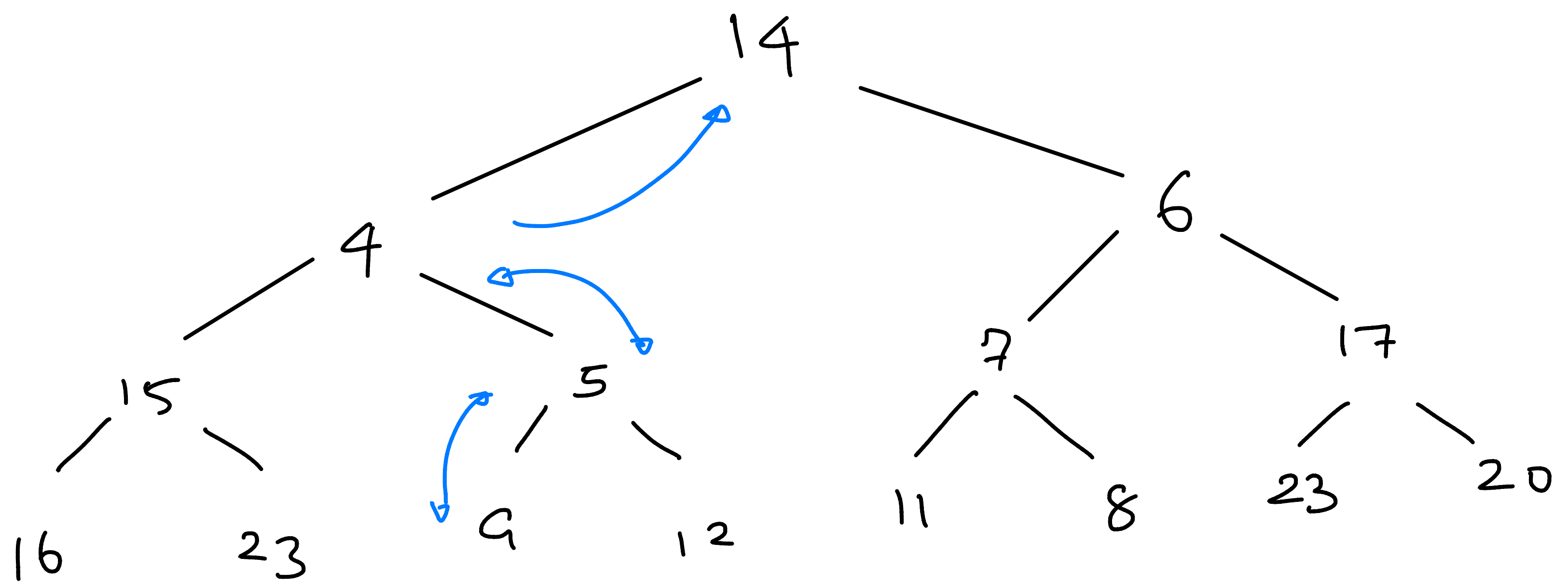


Building Heap from Bottom up:

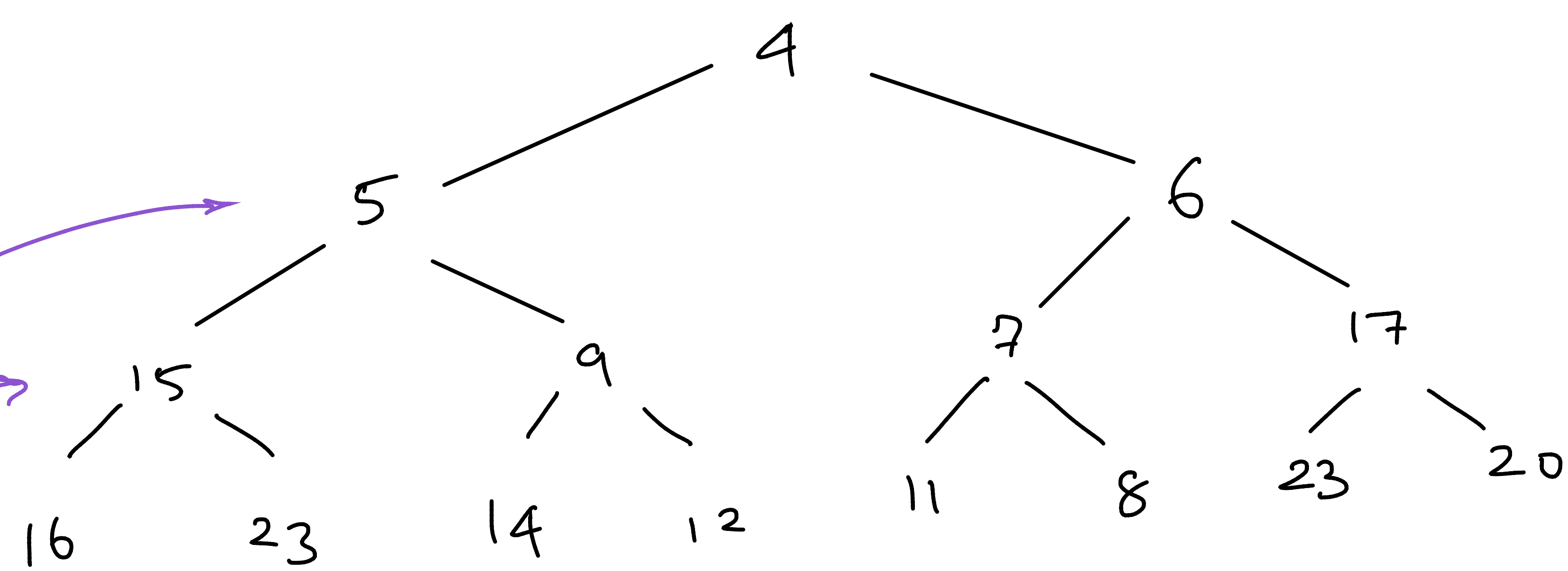




4
15
5
16
23
9
12
6
8
17
11
7
23
20



14
4
6
15
5
7
17
16
23
9
12
11
8
23
20



2^h nodes in last level   0 down heaps

2^{h-1} Downheap calls, each = 1 swap

2^{h-2} Downheap calls, each = 2 swaps

Total $\displaystyle\sum_{j=0}^{h} j\, 2^{h-j} = 2^h \underbrace{\sum_{j=0}^{h}\left(\frac{j}{2^j}\right)}_{\le 2} \le 2^{h+1}$

$\Rightarrow$ $O(n)$ to build the heap.

$$\sum_{j=0}^{h} \left(\frac{j}{2^j} \right) = \frac{1}{2} + \frac{2}{2^2} + \frac{3}{2^3} + \cdots$$

$$S = x + 2x^2 + 3x^3 + \cdots + h x^h$$

$$xS = x^2 + 2x^3 + \cdots + h x^{h+1}$$

$$S - xS = x + x^2 + x^3 + \cdots + x^h - h x^{h+1}$$

$$S(1-x) = x(1 + x + \cdots + x^{h-1}) - h x^{h+1}$$

$$S(1-x) = x \frac{(1-x^h)}{(1-x)} - h x^{h+1}$$

$$S = x \frac{(1-x^h)}{(1-x)^2} - \underbrace{\frac{h x^{h+1}}{(1-x)}}_{+ve} , \quad \text{put } x = \frac{1}{2}$$

$$S \leq x \frac{(1-x^h)}{(1-x)^2}$$

$$\frac{x\,(1-x^{n})}{(1-x)^2} = \frac{1}{2}\,\frac{(1-\frac{1}{2}x)}{(1-{}^1\!/_2)^2} \leq \frac{1}{2}\,\frac{1}{({}^1\!/_4)}$$

$$\leq 2$$