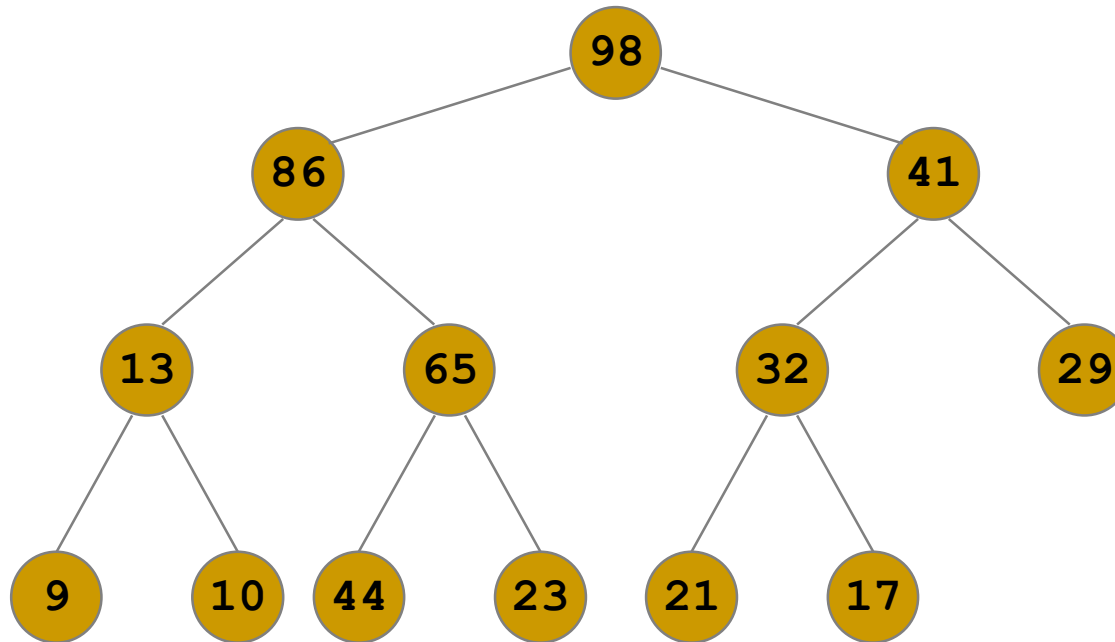# Using a Priority Queue

Partially Ordered Tree – max heap
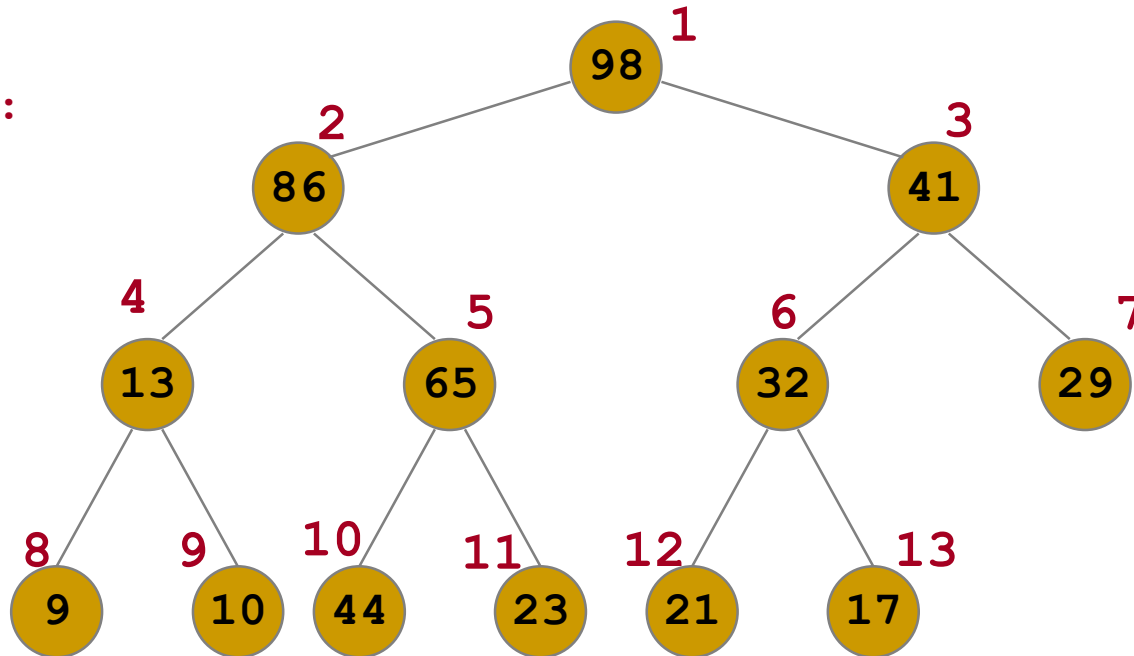

Note: an inorder traversal would result in:

9, 13, 10, 86, 44, 65, 23, 98, 21, 32, 17, 41, 29

Heap Array Example

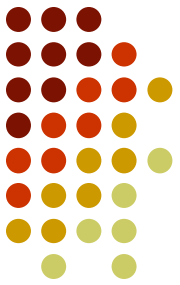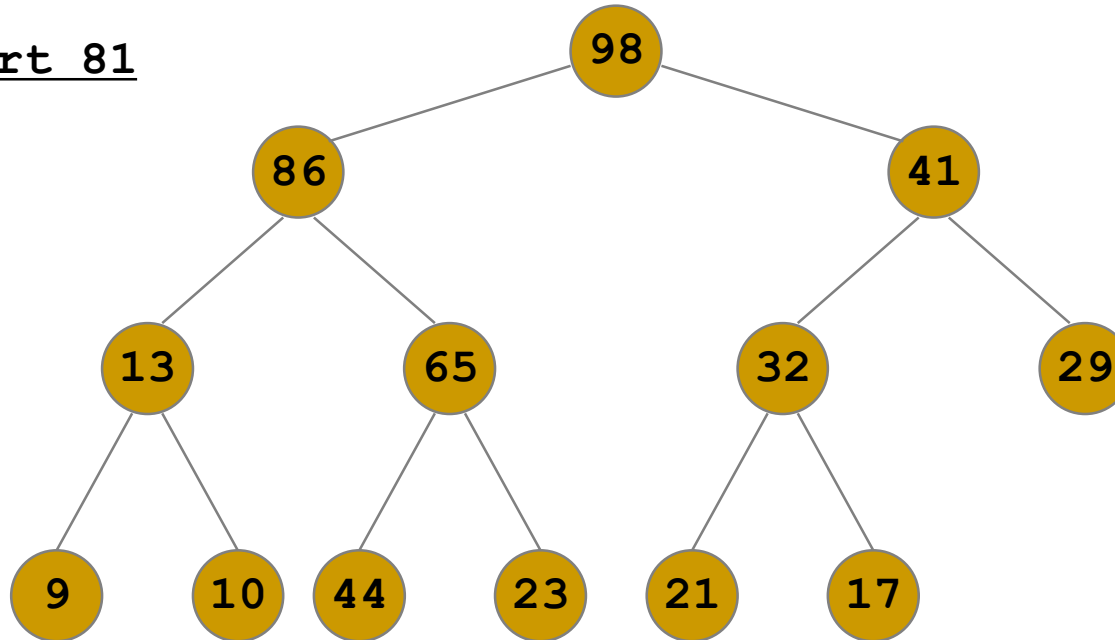**Heap:**

Underlying Array:

index	1	2	3	4	5	6	7	8	9	10	11	12	13
value	98	86	41	13	65	32	29	9	10	44	23	21	17

Heap Insertion Example

Insert 81

index	1	2	3	4	5	6	7	8	9	10	11	12	13	14
value	98	86	41	13	65	32	29	9	10	44	23	21	17	

Heap Insertion Example

Insert 81

81 is less than 98 so we are finished
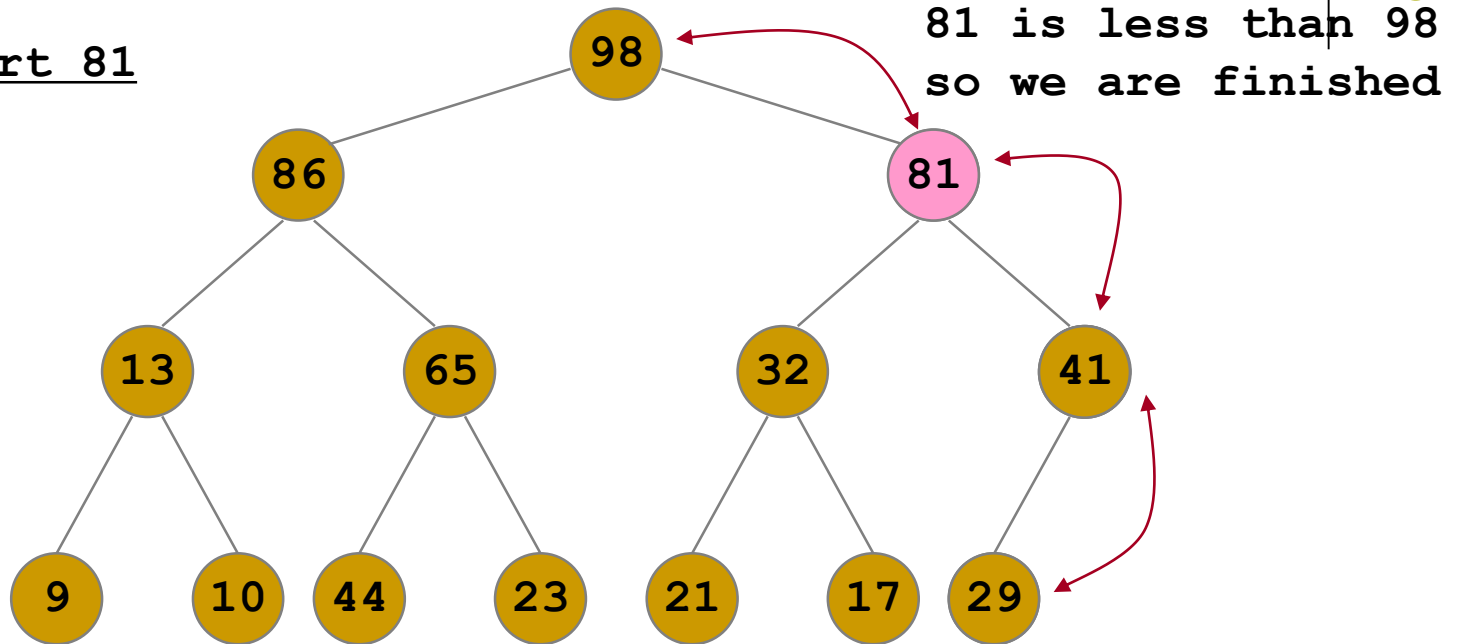
14/2 = 7

index	1	2	3	4	5	6	7	8	9	10	11	12	13	14
value	98	86	81	13	65	32	41	9	10	44	23	21	17	29

Heap Removal Example
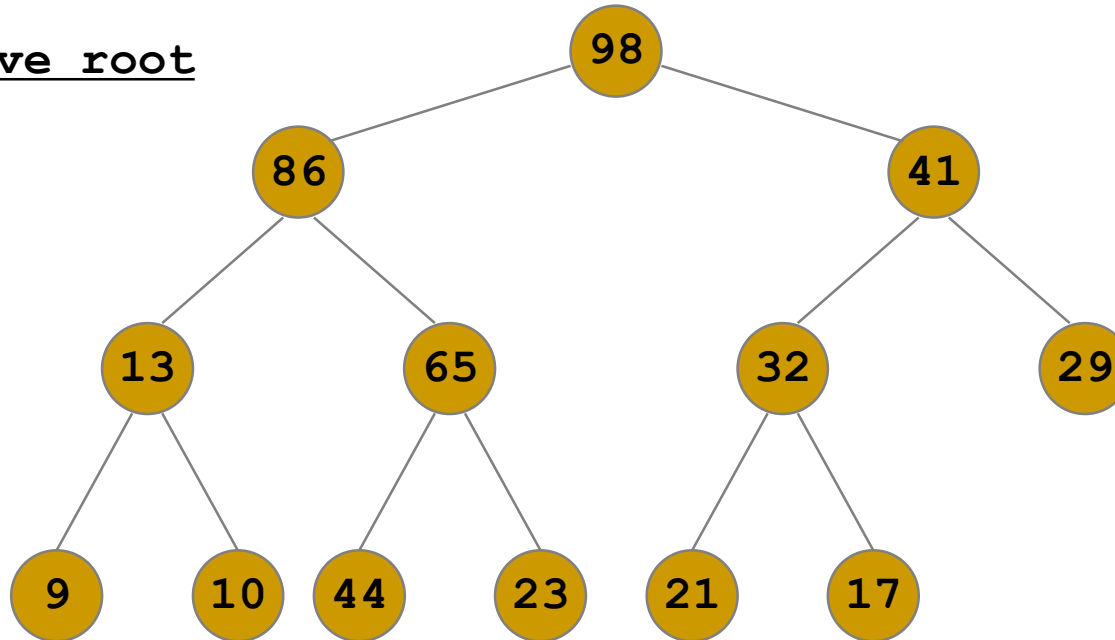
Remove root

index	1	2	3	4	5	6	7	8	9	10	11	12	13
value	98	86	41	13	65	32	29	9	10	44	23	21	17

Heap Removal Example

Remove root

86

? → → ← **?**
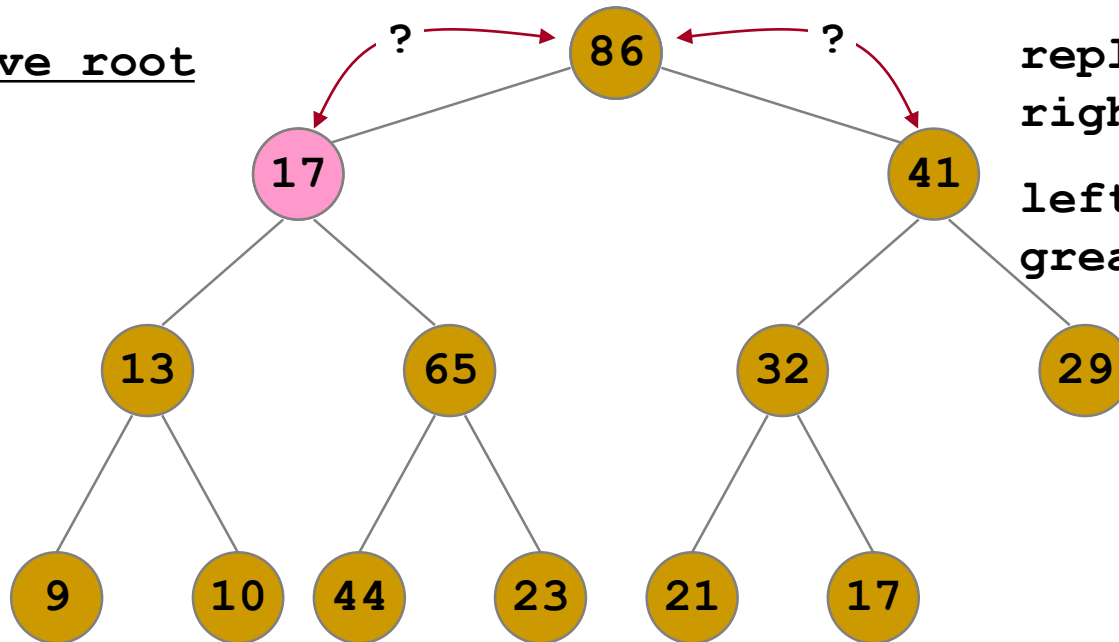
replace root with right-most leaf

left child is greater

17 **41**

13 **65** **32** **29**

9 **10** **44** **23** **21** **17**

children of root: 2*1, 2*1+1 = 2, 3

index	1	2	3	4	5	6	7	8	9	10	11	12	13
value	86	17	41	13	65	32	29	9	10	44	23	21	

Heap Removal Example

Remove root

right child is greater

86

? → 65 ← ?

13   17   41

9   10   44   23   32   29

21
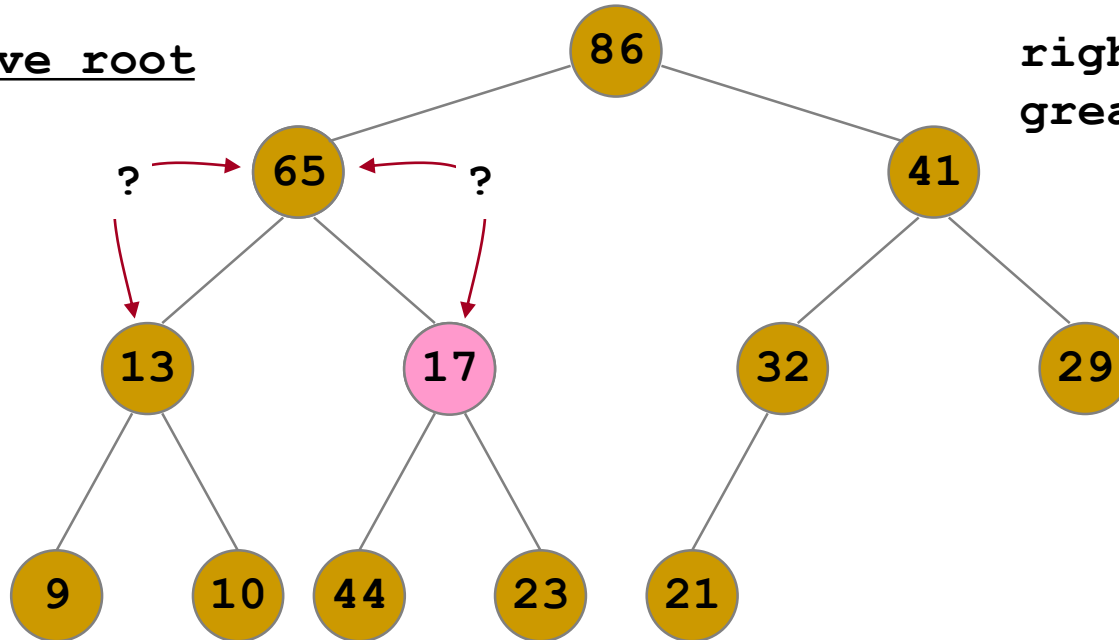
children: 2*2, 2*2+1 = 4, 5

index	1	2	3	4	5	6	7	8	9	10	11	12	13
value	86	65	41	13	17	32	29	9	10	44	23	21	

Heap Removal Example

Remove root

left child is greater

86

65   41

13   44   32   29

?   ?

9   10   **17**   23   21

98

86   41

13   65   32   29

9 10 44 23   21 17
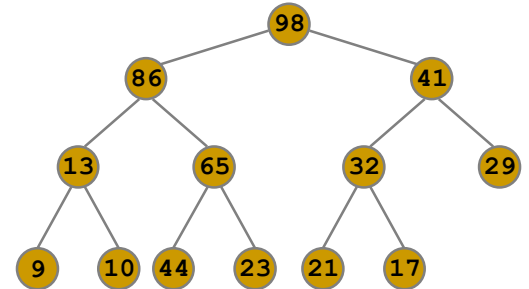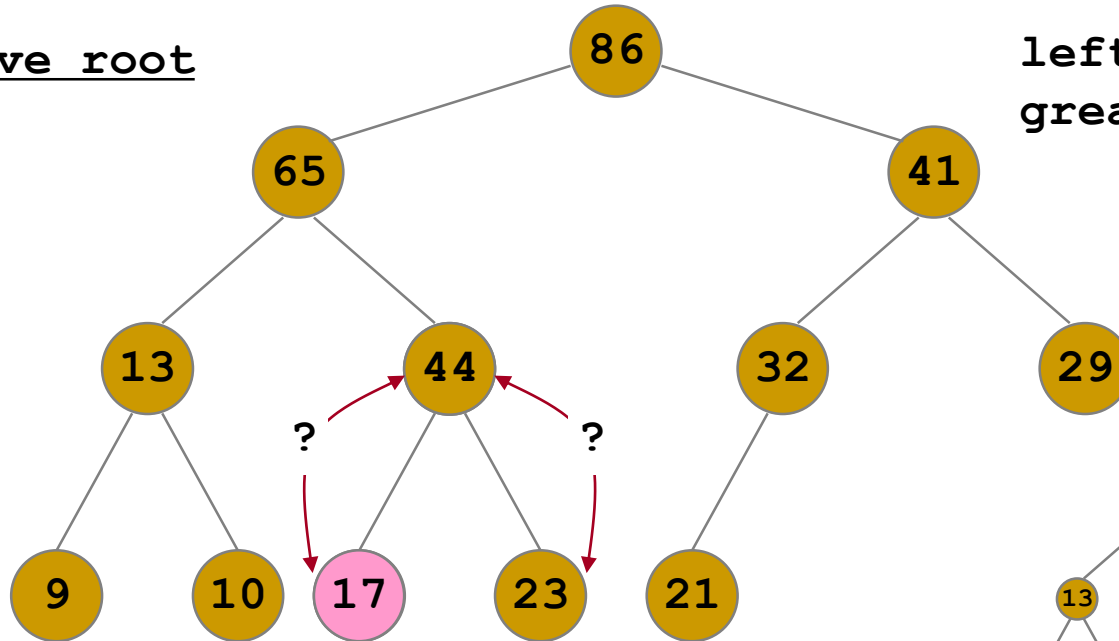
children: 2*5, 2*5+1 = 10, 11

index	1	2	3	4	5	6	7	8	9	10	11	12	13
value	86	65	41	13	44	32	29	9	10	17	23	21	