2018SS

Question 49

[1.5 marks] What is the output of the following code? The array-based list representing the binary heap (the `bin_heap` attribute) will be printed by the print statement.

```
pq = PriorityQueue()
pq.insert(24)
pq.insert(10)
pq.insert(7)
pq.insert(99)
pq.insert(11)
print(pq)
```

(a) `[0, 7, 11, 10, 99, 24]`
(b) `[7, 10, 11, 24, 99]`
(c) `[0, 24, 10, 7, 99, 11]`
(d) `[0, 7, 10, 11, 24, 99]`
(e) None of the above

Question 50

[1.5 marks] What is the output of the following code? The array-based list representing the binary heap (the `bin_heap` attribute) will be printed by the print statement.

```
pq = PriorityQueue()
pq.insert(24)
pq.insert(10)
pq.insert(7)
pq.insert(99)
pq.insert(11)
pq.del_min()
pq.del_min()
print(pq)
```

(a) `[11, 24, 10]`
(b) `[0, 11, 99, 24]`
(c) `[0, 10, 11, 24, 99]`
(d) `[0, 11, 24, 99]`
(e) None of the above

Question 51

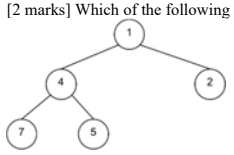
[1 mark] What is the Big-O time complexity for inserting an item into a priority queue if it is implemented using a Binary Heap data structure?

(a) O(n log n)
(b) $O(n^2)$
(c) O(n)
(d) O(log n)
(e) None of the above

2017 S2

Question 30

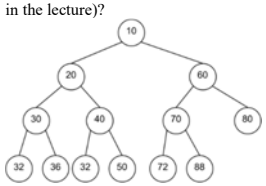
[2 marks] Which of the following insertion sequences will produce the binary heap shown below?

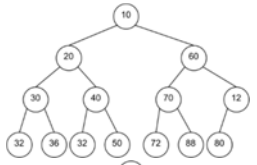


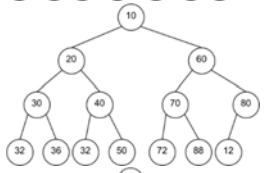
(a) 1 5 2 7 4
(b) 4 7 2 5 1
(c) 5 1 4 2 7
(d) 7 4 2 5 1
(e) None of the above.

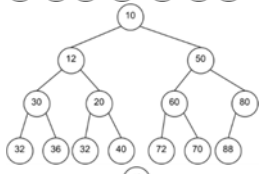
Question 34

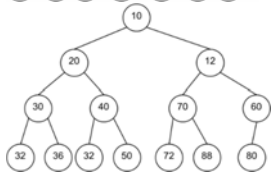
[2 marks] Given is the binary heap in the picture below. What does the heap look like after inserting the value 12 into it (using the method presented in the lecture)?



(a)


(b)


(c)


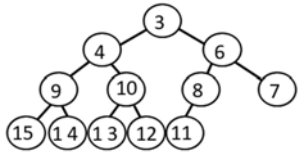
(d)


(e) None of the above.

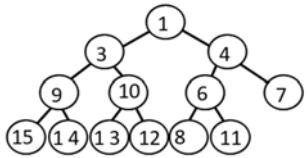
2017 S1

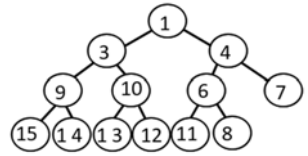
Question 50

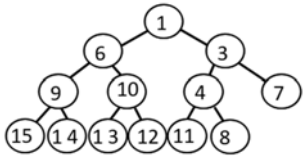
[2 marks] Given the binary heap in the picture below:

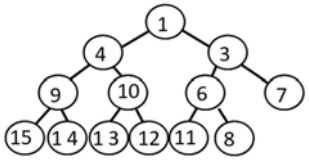


what does the heap look like after inserting the value **1** into it (using the method presented in lectures)?

(a) 

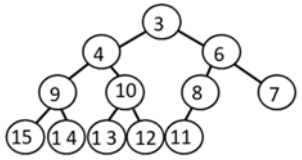
(b) 

(c) 

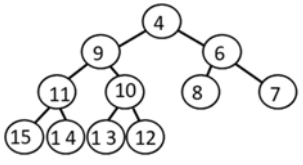
(d) 

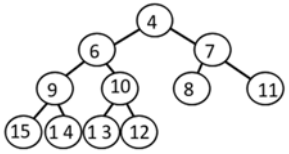
(e) None of the above

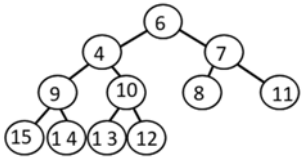
Question 51
[2 marks] Given the binary heap in the picture below:

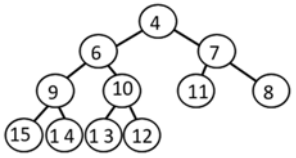


what does the heap look like after removing its minimum value (using the method presented in lectures)?

(a) 

(b) 

(c) 

(d) 

(e) None of the above

2017 SS
Question 37 **[8 marks]**

The initialisation function and the `__str__()` function for a `PriorityQueue` class are shown below. This class uses a **min** binary heap, stored in a Python list, to represent the priority queue.

```
class PriorityQueue:

    def __init__(self):
        self.bin_heap = [0]
        self.current_size = 0

    def __str__(self):
        return str(self.current_size) + " : " + str(self.bin_heap[1:])
```

A `PriorityQueue` object, `pq`, has been created and has had the following **seven** values inserted, in this order:

40 , 20 , 30 , 50 , 10 , 5 , 8

The code for performing these insertions is as follows:

```
pq = PriorityQueue()
pq.insert(40)
pq.insert(20)
pq.insert(30)
pq.insert(50)
pq.insert(10)
pq.insert(5)
pq.insert(8)
```

What is the output ***after*** the following code is executed, in the order shown below (note the first part of the output string - the size of the priority queue - is already shown)?

a) printing the current priority queue `print(pq)`	7 : [ ]
b) then, continuing *after* (a), inserting the value 15 and then printing the resulting priority queue:	8 : [ ]

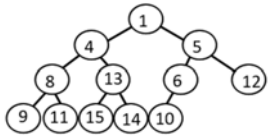
`pq.insert(15)` `print(pq)`	

c) finally, continuing *after* (b), deleting the minimum value from the priority queue, and then printing the remaining values:
 	
`pq.del_min()`
`print(pq)`
	7 : [ ]

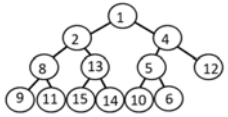
2016 S2

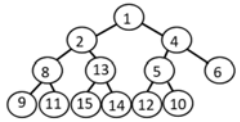
Question 45

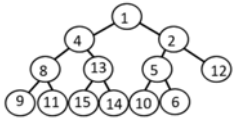
[2 marks] Given is the binary heap in the picture below.

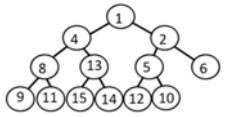


What does the heap look like after inserting the value 2 into it (using the method presented in lectures)?

(a)


(b)


(c)


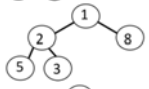
(d)


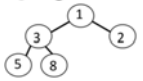
(e) None of the other answers.

Question 46

[2 marks] A heap can be created from a list by inserting the list values one by one. This results in an O(n log n) algorithm. In lectures we discussed a more efficient O(n) algorithm, which we used in the init method of the `BinHeap` class. Given is a list [5, 3, 1, 8, 2]. What does the resulting heap look like when constructing it with the O(n) method discussed in lectures?

(a)


(b)


(c)


(d)


(e) None of the other answers.

2016S1

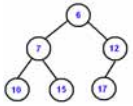
Question 21

[1 mark] Which of the following data structures provides the most computationally efficient implementation of a priority queue?

(a) binary heap
(b) sorted list
(c) binary search tree
(d) queue
(e) None of the above

Question 25

[1 mark] Which of the following lists represents the binary heap below?



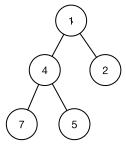
(a) [0, 6, 7, 12, 10, 15, 17]
(b) [0, 17, 15, 10, 12, 7, 6]
(c) [0, 6, 7, 10, 12, 15, 17]
(d) [6, 7, 12, 10, 15, 17]
(e) None of the above

Question 27

[1 mark] When searching for a key value which is present in a binary heap tree with exactly 15 nodes, what is the **maximum** number of comparisons needed?

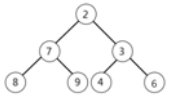
(a) 3
(b) 8
(c) 15
(d) 4
(e) None of the above

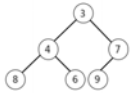
Question 28
[1.5 marks] Which of the following insertion sequences will produce the binary heap shown below?

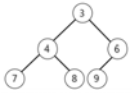


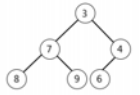
(a) 4 7 2 5 1
(b) 7 4 2 5 1
(c) 5 1 4 2 7
(d) 1 5 2 7 4
(e) None of the above

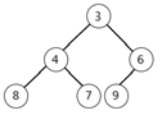
Question 32
[1.5 marks] Given the heap below. What does the heap look like after **removing** the **root** element?



(a) 

(b) 

(c) 

(d) 

(e) None of the above

2016 SS
Question 30
A *priority queue* can be implemented with various data structures, such as an unsorted Python list, a sorted linked list, and a binary heap. Which of the following is the correct order, from most efficient to least efficient, for implementing the "**remove min/max**" operation using these data structures?

(a) sorted linked list, unsorted Python list, binary heap
(b) binary heap, unsorted Python list, sorted linked list
(c) binary heap, sorted linked list, unsorted Python list
(d) sorted linked list, binary heap, unsorted Python list
(e) unsorted Python list, binary heap, sorted linked list

Question 37 **[9 marks]**

The initialisation function and the __str__() function for a PriorityQueue class are shown below. This class uses a **min** binary heap (i.e. the smallest value at the root), stored in a Python list, to represent the priority queue.

```
class PriorityQueue:
```



```
    def __init__(self):
        self.bin_heap = [0]
        self.current_size = 0

    def __str__(self):
        return str(self.current_size) + " : " + str(self.bin_heap[1:])
```

A PriorityQueue object, pq, has been created and has had the following six values inserted, in this order:

30, 20, 15, 12, 13, 18

The code for performing these insertions is as follows:

```
pq = PriorityQueue()
pq.insert(30)
pq.insert(20)
pq.insert(15)
pq.insert(12)
pq.insert(13)
pq.insert(18)
```

What is the output after each of the following statement are executed, in the order shown below (note that in each question, the first part of the output string - the size of the priority queue - is already shown)?

a) printing the current priority queue `print(pq)`	**`6 :`**

b) then, continuing after (a), inserting the value 17 and printing the resulting priority queue:
`pq.insert(17)`
`print(pq)`
	`7 :`

c) finally, continuing after (b), deleting the minimum value from the priority queue, and then printing the remaining values:
`pq.del_min()`
`print(pq)`
	`6 :`