IE - Data Structures and Algorithms - 2020 - 2021

Home / My courses / IE - DSA / Quiz / Lecture 8 Quiz

Quiz navigation

1 ● 2 ● 3 ✓ 4 ✓ 5 ✓ 6 ✓ 7 🟥 8 ✓ 9 🟥 10 ✓ 11 🟥 12 ✓

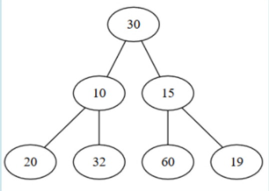
Show one page at a time

Finish review

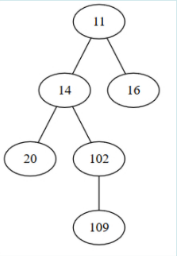
Started on	Sunday, 18 April 2021, 10:29 PM
State	Finished
Completed on	Sunday, 18 April 2021, 10:44 PM
Time taken	15 mins 2 secs
Marks	8.42/12.00
Grade	**7.01** out of 10.00 (**70**%)

Question **1**
Partially correct
Mark 0.75 out of 1.00
Flag question

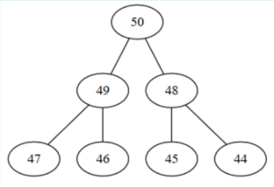
Match the drawings to the description.



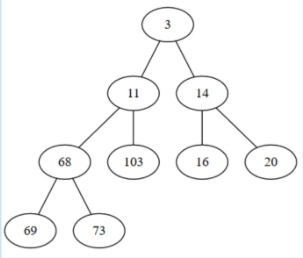
It does have a heap structure, but it does not respect the heap property ✓



It has neither a heap structure, nor a heap property. ✗



It is a valid max-heap. ✓

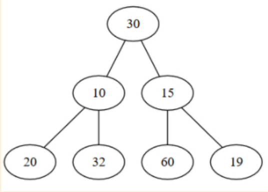


It is a valid min-heap. ✓

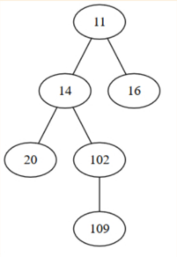
Your answer is partially correct.

You have correctly selected 3.

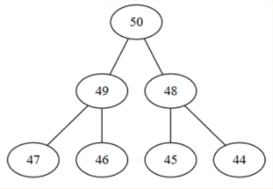
The correct answer is:



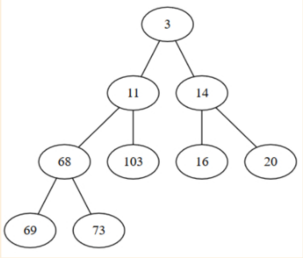
→ It does have a heap structure, but it does not respect the heap property,



→ It does not have a heap structure, but it respects the heap property.,



→ It is a valid max-heap.,



→ It is a valid min-heap.

Question **2**
Partially correct
Mark 0.67 out of 1.00
Flag question

Which of the following arrays represents a valid (min or max) heap?

Select one or more:

☑ 41 60 90 95 100 97 108 100 121 101 ✓ It is a valid min-heap

☐ 2 43 90 66 86 95 95

☑ 65 58 15 50 46 6 8 38 34 41 ✓ It is a valid max-heap

□ 69 59 55 45 87 22 7 4 22 52 32

□ 53 16 38 7 10 2 15 8 5

Your answer is partially correct.

You have correctly selected 2.
You need to draw them in the tree form and check if it respects the property.

The correct answers are: 2 43 90 66 86 95 95, 65 58 15 50 46 6 8 38 34 41, 41 60 90 95 100 97 108 100 121 101

Question **3**
Correct
Mark 1.00 out of 1.00
Flag question

Assume that the following array represents a binary max-heap. How many elements do not respect the heap property (count an element if its child/children are not respecting the property. If a node has no child, it is considered to respect the property).

[21, 2, 45, 37, 38, 28, 24, 45, 1, 20, 63]

Select one or more:

□ 6

□ 5

☑ 4 ✓

□ 1

□ 8

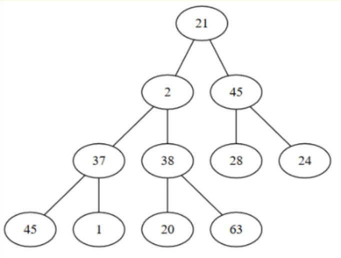
□ 7

□ 3

□ 2

Your answer is correct.

This is how the array looks like in the tree form:

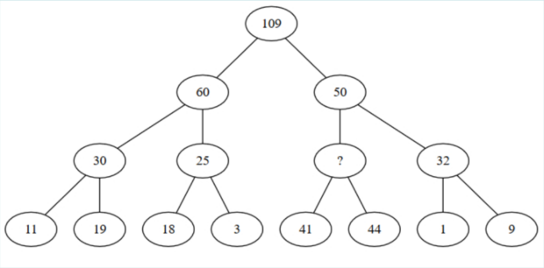


In a max heap every node has to be greater than or equal to its children. Nodes not respecting this property are: 21, 2, 37 and 38.

The correct answer is: 4

Question **4**
Correct
Mark 1.00 out of 1.00
Flag question

Consider the following binary heap that contains unique values:



Which one(s) of the following values can be in place of "?" ?

Select one or more:

□ 30

□ 52

□ 35

□ 40

☑ 49 ✓

□ 41

□ 10

☑ 47 ✓

Your answer is correct.

It is a max heap. Heap property tells us that every nodes has to be greater than its children. So ? has to be less than its parent (50) and greater than its children (41 and 44). Possible values are (considering unique elements): 45, 46, 47, 48, 49.

The correct answers are: 47, 49

Question **5**
Correct
Mark 1.00 out of 1.00
Flag question

Assume you have a max-heap with 19 elements. How many levels does the heap have?

Select one or more:

□ 2

☑ 5 ✓

□ 1

□ 4

□ 6

□ 7

□ 8

□ 3

Your answer is correct.

We have the nodes on levels like this: 1 + 2 + 4 + 8 + 4 (last level is not complete). So 5 levels.

The correct answer is: 5

Question **6**

Correct

Mark 1.00 out of 1.00

Flag question

Assume you have the following min-heap.

[19, 60, 20, 80, 101, 25, 40, 99, 103]

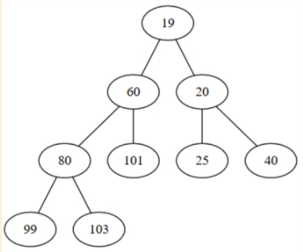
How is the heap going to look like after adding the element 23?

Select one or more:

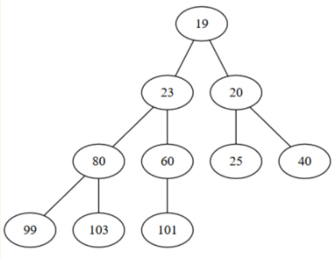
- □ [19, 20, 23, 80, 60, 101, 25, 40, 99, 103]
- □ [19, 60, 20, 23, 80, 101, 25, 40, 99, 103]
- □ [19, 60, 20, 80, 101, 25, 40, 99, 23, 103]
- □ [19, 60, 20, 80, 101, 25, 40, 99, 103, 23]
- ☑ [19, 23 20, 80, 60, 25, 40, 99, 103, 101] ✓

Your answer is correct.

Initially the heap looks like this (it is easier if we work with the tree representation):



Initially we put 23 as the left child of 101 and then we bubble it up. We will need to swap it with 101 and 60, so the final result will be:



The correct answer is: [19, 23 20, 80, 60, 25, 40, 99, 103, 101]

Question **7**

Incorrect

Mark 0.00 out of 1.00

Flag question

Consider the following heap:

[19, 60, 20, 80, 101, 25, 40, 99, 103]

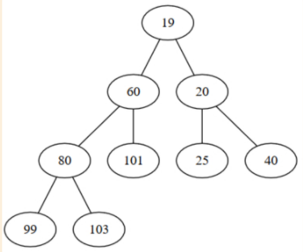
How will the heap look like if we remove an element?

Select one or more:

- ☑ [19, 60, 20, 80, 101, 25, 40, 99] ✗
- ☑ [20, 60, 25, 80, 101, 40, 99, 103] ✗
- □ [103, 60, 20, 80, 101, 25, 40, 99]
- □ [20, 60, 25, 80, 101, 103, 40, 99]
- □ [20, 25, 60, 80, 101, 40, 99, 103]

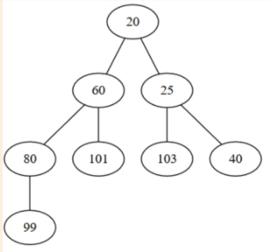
Your answer is incorrect.

Initially the heap looks like this (it is easier to work with the tree format)



:

We can only remove the root. We move 103 in place of the root and bubble it down. It will change place with 20 and 25 (remember, we always compare it with the smallest child).



The correct answer is: [20, 60, 25, 80, 101, 103, 40, 99]

Question **8**

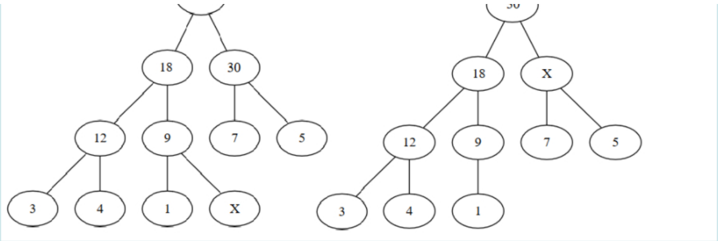
Correct

Mark 1.00 out of

From the heap on the left (which contains unique elements) we have removed an element. The result is the heap on the right. What value(s) can be in the node with X?

1.00

Flag question



Select one or more:

☐ 12

☑ 8 ✓

☐ 2

☐ 20

☐ 1

Your answer is correct.

It is a max-heap, every node is greater than its children. From the figure on the left we know that X is less than 9.

When we remove something, 50 is going to be removed and X will be moved to the root. Then we start bubble-down, we compare X with the maximum child, 30. We know from the result that X was swapped with 30, so 30 > X. Then, still in the bubble-down process, we compare X with the maximum of the new children, 7. Since X was not swapped with 7, we know that it must be greater than 7.

So we have:

X < 9

X > 7

X has to be 8.

The correct answer is: 8

Question **9**

Incorrect

Mark 0.00 out of 1.00

Flag question

Assume you have the following heap:

[13, 25, 19, 60, 28, 40, 50, 77, 80, 30, 29, 41]

How will the array look like after three delete operations?

Select one or more:

☐ 28 40 29 30 60 50 77 80 41

☐ 28 29 40 60 30 41 50 77 80

☐ 60 28 40 50 77 80 30 29 41

☐ 60 28 40 50 77 80 30 29 41

☑ 13 25 19 60 28 40 50 77 80 ✗

Your answer is incorrect.

Remember, you can only delete the root. You move the "last" element to replace the root and after that do a bubble-down.

The correct answer is: 28 29 40 60 30 41 50 77 80

Question **10**

Correct

Mark 1.00 out of 1.00

Flag question

Assume that I have a max-heap. What is the complexity of finding the maximum element from the heap?

Select one or more:

☐ Θ(log2 n)

☐ Θ(n)

☐ O(n)

☑ Θ(1) ✓

☐ O(log2 n)

Your answer is correct.

In a max-heap the root is the maximum. It is on position 1.

The correct answer is: Θ(1)

Question **11**

Incorrect

Mark 0.00 out of 1.00

Flag question

Assume you have a max-heap. What is the complexity of finding the minimum element of the heap?

Select one or more:

☐ Θ(n/2)

☑ Θ(log2n) ✗

☐ Θ(1)

☐ O(n)

☐ Θ(n)

Your answer is incorrect.

In a max-heap every node is greater than its children. So the minimum must not have a child, it has to be a leaf node. The last element of the heap is on position n. Its parent is on position n/2. This is the last node with a child. Nodes between positions n/2+1 and n are leaf nodes. Any of them can be the minimum. We need to check aproimately n/2 nodes. As a complexity this is still Θ(n) (2 is just a constant and we can ignore it)

The correct answer is: Θ(n)

Question **12**

Correct

Mark 1.00 out of 1.00

Flag question

In a max-heap what is the complexity of finding the second largest element?

Select one or more:

☐ O(n)

☐ O(log2n)

☑ Θ(1) ✓

☐ Θ(n)

☐ Θ(log2n)

Your answer is correct.

In a max-heap the root is the maximum. The second largest element has to be a child of the maximum, so it can only be on position 2 or 3. It means that we only need to check a constant number of positions.

The correct answer is: Θ(1)

Finish review

◄ Lecture 7 Quiz

Jump to...