FIT1045 MCQ

Test your understanding!

Is the code below correct?

```
# Assume N>1. Returns True if N is prime, False otherwise
def isPrime(N):
    for i in range(2,N):
        if N%i == 0:
            return False
        else:
            return True
```

A. Yes
B. No
C. No idea

The following brute force algorithm considers numbers from 2 to N-1. Will the algorithm be still correct if we consider only the numbers between 2 to square root of N (inclusive)?

```
# returns True if N is prime, otherwise returns False
def isPrime(N):
    for i in range(2,N):
        if N%i == 0:
            return False
    return True
```

A. Yes
B. No
C. No idea

Consider the string Str = “**agttacgatta**” then “**tta**” is the same as:

A) Str[2:5]

B) Str[3:6]

C) Str[8:11]

D) Both A and C

What is the output of this code?

```
def getMinIndex(myList, start, stop):
    min_index = start
    for i in range(start+1,stop):
        if myList[i] < myList[min_index]:
            min_index = i
    return min_index

aList = [1,11,13,8,4,2,5,6]
n = len(aList)
min_position = getMinIndex(aList,3,n)
print(min_position)
```

A. 2

B. 5

C. 1

D. None of the above

What will be the output of this code?

```
aList=[2,4,6,8,5]
#swap elements at index 3 and index 4
aList[3] = aList[4]
aList[4] = aList[3]
print(aList)
```

A. **[2, 4, 6, 5, 8]**

B. **[2, 4, 6, 5, 5]**

C. **[2, 4, 5, 6, 8]**

D. **None of the above**

What does the list [5, 3, 1, 2, 4] contain after the first iteration of the outer loop in Selection sort?

A. [5, 3, 1, 2, 4]

B. [3, 5, 1, 2, 4]

C. [1, 3, 5, 2, 4]

D. [1, 3, 1, 2, 4]

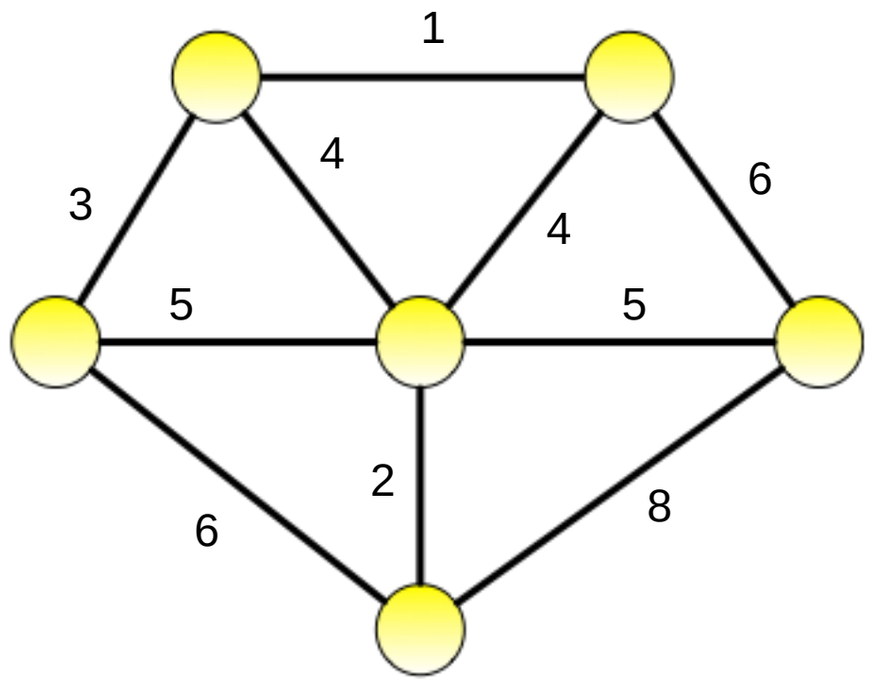
What is the total weight of the minimum spanning tree for the following graph?

A) 14

B) 15

C) 16

D) None of the above



Using 1c, 5c, and 10c coins, what is the minimum number of coins required to give change of 12 cents?

A) 2

B) 3

C) 4

D) None

Does this greedy algorithm always return optimal results?

A. Yes

B. No

C. Not sure

Suppose you have coins of values 1, 5, 6, and 9 cents. What is the minimum number of coins required to give a change of 12 cents?

A. 4

B. 1

C. 2

D. 3

If you can only carry 20kg, what is the maximum of total value of items you can carry?

Item	1	2	3	4	5	6
Weight	20kg	10kg	9kg	4kg	2kg	1kg
Value	$4000	$3500	$1800	$400	$1000	$200

A) $4000

B) $4900

C) $5100

D) $5000

E) None of the above

Consider the code:

```
x = [4, 0.5,-10,2]
y = x
x[0] = 10
print(y)
```

What will be printed?

A. [4, 0.5, -10, 2]
B. [10, 0.5, -10, 2]
C. [10]
D. None of the above

Consider the code:

```
x = [1,2]
print(id(x))
x = x*2
print(id(x))
```

Will the two print statements print the same value?

A. Yes
B. No
C. No idea

Consider the code:

```
x = [1,2]
y = []
y.append(x)
y.append(x)
y[0][0] = 4
print(y)
```

What will be printed?

A. [[1,2],[1,2]]
B. [[4,2],[1,2]]
C. [[4,2],[4,2]]
D. None of the above

Consider the code:

```
x = [1,2]
y = []
y.append(x[:])
y.append(x[:])
y[0][0] = 4
print(y)
```

What will be printed?

A. [[1,2],[1,2]]
B. [[4,2],[1,2]]
C. [[4,2],[4,2]]
D. None of the above

What will be printed?

```
def double(val):
    val = val*2

x = [4,3]
double(x)
print(x)
```

Note we are NOT changing in-pla

What will be printed?

A. [4,3]
B. [4,3,4,3]
C. Not sure

Suppose your friend has 9 coins which all appear identical. One is known to be ***lighter*** than the rest.

What is the minimum number of weighings required to guarantee that you find it?

A. 1
B. 2
C. 3
D. None of the above

Which subset of

{‘ape’, ‘dog’, ‘cat’, ‘snake’, ‘fish’, ‘bird’}

does the following bit list [1, 0, 1, 0, 0, 1] represent?

A. {‘dog’, ‘snake’, ‘bird’}
B. {‘ape’, ‘cat’, ‘bird’}
C. {‘ape’, ‘cat’, ‘fish’}
D. {} – the empty set

Aside: Swapping variables

After the following what are the values of a and b?

```
a = 20
b = 50
a = b
b = a
```

A. a=50,b=50

B. a=20,b=50

C. a=50,b=20

D. a=20,b=20

Suppose you have the following function:

```
def double(val):
    val *= 2
```

What is printed by the following code?

```
>>> val = [‘a’, ‘b’, ‘c’]
>>> double(val)
>>> val
```

A. [‘a’, ‘b’, ‘c’]
B. [‘a’, ‘b’, ‘c’, ‘a’, ‘b’, ‘c’]
C. None of the above

Suppose you have the following function:

```
def double(k):
    k = 2*k
    return k
```

What is printed by the following code?

```
>>> k = 10
>>> k = double(k)
>>> k
```

A. 10
B. 20
C. None of the above

Given the code below. How many times is the power function called including the call to power(2,14)?

```
def Power(x, N):
    if N == 0:
        return 1
    else:
        value = Power(x, N//2)
        if N % 2 == 0:
            value = value*value
        else:
            value = value*value*x
        return value
print(Power(2,14))
```

A. 1
B. 4
C. 5
D. 6
E. 7
F. ☹ ☹ ☹

Check your understanding

Assuming a stack started off with the following items in it

What would be at the top of the stack after the following operations?

push 10, push 2, push 5, pop pop

A. 4
B. 8
C. 10
D. 2
E. 5

Implementing stacks…

Which of the following statements are true

Given the list: L = [1,65,3,6]

I) L.append(4) gives [1,65,3,6,4]

II) L.append(4) gives [4,1,65,3,6]

III) L.pop() leaves L as [1,65,3]

IV) L.pop() leaves L as [65,3,6]

A. I and III are true
B. I and IV are true
C. II and III are true
D. II and IV are true

Checking understanding

Assuming a queue with the following contents

front 

rear

what is the size of this queue after the following operations:

Serve, serve, append 20, serve, append 4

A. 2
B. 3
C. 4
D. 5
E. 6
F. 7

Removing negative numbers from a queue

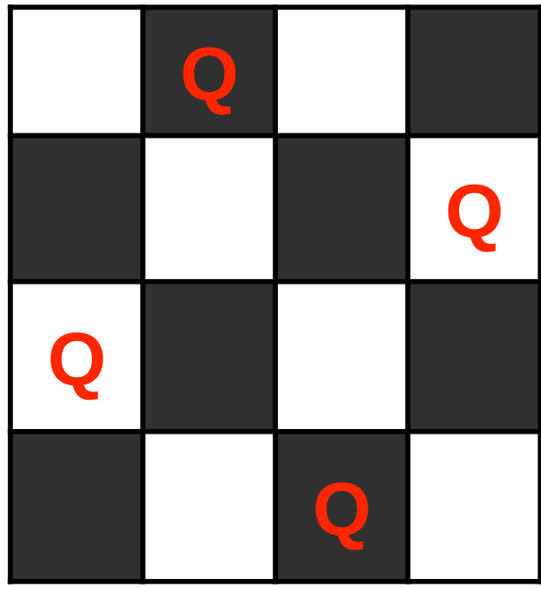
Let's assume we have a queue of numbers like so:

If we wanted to remove all negative numbers from the queue, what must replace the "???"

```
"""
This function will take a queue of numbers and remove the items < 0
"""
def removeNegatives(aQueue):
    numberSeen = 0
    n = len(aQueue)
    while numberSeen <= n:
        item = aQueue.pop(0)
        numberSeen+=1
        if item ???:
            aQueue.append(item)
```

A. < 0
B. >= 0
C. == stopValue
D. *None of the above*

In the 4 queens problem, the board



A. is a solution, represented by [2,0,3,1]

B. is a solution, represented by [3,1,4,2]

C. is represented by [2,0,3,1], and is not a solution.

D. None of the above

For the 8-Queens problem, a partial solution is a solution if its length is 8.

A. True

B. False

Of the following lists, which ones represent partial solutions for the Knapsack problem:

Item	1	2	3	4	5	6
Weight	20kg	10kg	9kg	4kg	2kg	1kg
Value	$4000	$3500	$1800	$400	$1000	$200

list1 = [1, 2]

list2 = [1]

list3 = [2,3,4]

list4 = [5, 6]

A. list1 and list2
B. list1 and list3
C. list2 and list4
D. None of the above

Suppose [3,6] represents the partial solution. Which of the following represents a list of possible items?

Item	1	2	3	4	5	6
Weight	20kg	10kg	9kg	4kg	2kg	1kg
Value	$4000	$3500	$1800	$400	$1000	$200

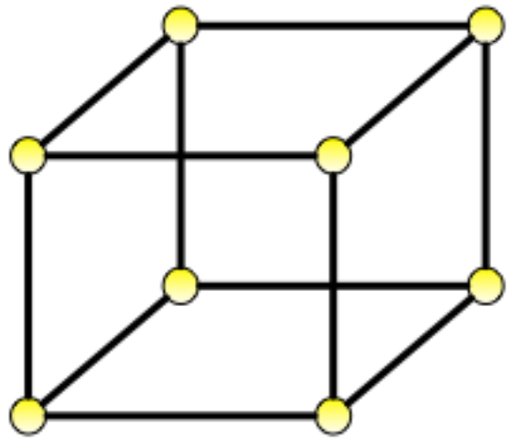
list1 = [2,4,5]

list2 = [4,5]

list3 = [3,4,5]

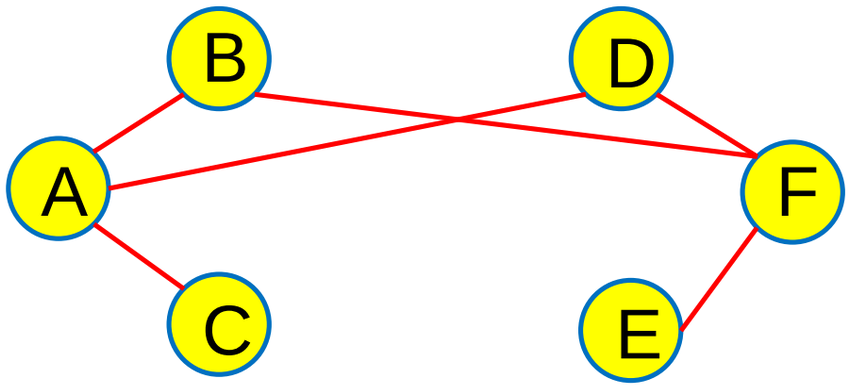
A. list1
B. list2
C. list3
D. None of the above

The following graph has a Hamiltonian cycle.



A. True
B. False

There is a vertex cover of size 2 or less for the following graph.



A. True

B. False

Given the code below. How many times is the power function called including the call to power(2,14)?

```
def Power(x, N):
    if N == 0:
        return 1
    else:
        value = Power(x, N//2)
        if N % 2 == 0:
            value = value*value
        else:
            value = value*value*x
        return value
print(Power(2,14))
```

A. 1
B. 4
C. 5
D. 6
E. 7
F. ☹ ☹ ☹

Check your understanding

Assuming a stack started off with the following items in it

What would be at the top of the stack after the following operations?

push 10, push 2, push 5, pop pop

A. 4
B. 8
C. 10
D. 2
E. 5

Implementing stacks…

Which of the following statements are true

Given the list: L = [1,65,3,6]

I) L.append(4) gives [1,65,3,6,4]

II) L.append(4) gives [4,1,65,3,6]

III) L.pop() leaves L as [1,65,3]

IV) L.pop() leaves L as [65,3,6]

A. I and III are true
B. I and IV are true
C. II and III are true
D. II and IV are true

Checking understanding

Assuming a queue with the following contents

front 

rear

what is the size of this queue after the following operations:

Serve, serve, append 20, serve, append 4

A. 2
B. 3
C. 4
D. 5
E. 6
F. 7

Removing negative numbers from a queue

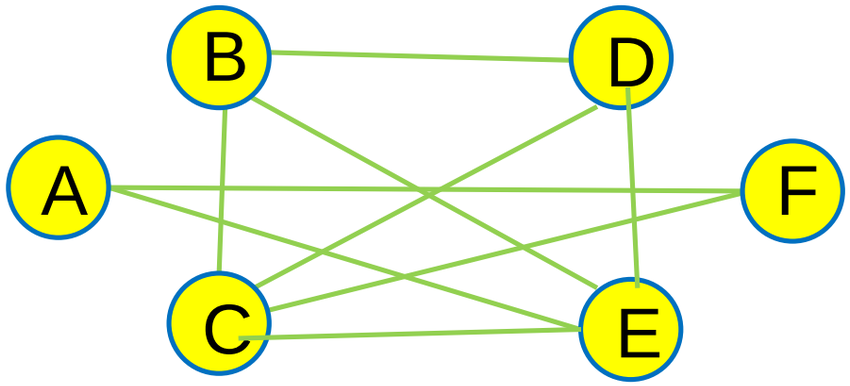
Let's assume we have a queue of numbers like so:

If we wanted to remove all negative numbers from the queue, what must replace the "???"

```
"""
This function will take a queue of numbers and remove the items < 0
"""
def removeNegatives(aQueue):
    numberSeen = 0
    n = len(aQueue)
    while numberSeen <= n:
        item = aQueue.pop(0)
        numberSeen+=1
        if item ???:
            aQueue.append(item)
```

A. < 0
B. >= 0
C. == stopValue
D. *None of the above*

There is a clique of four vertices.



A. True

B. False

For the input n=5, the number of integers the following function prints is:

```
def collatz(n):
    while n > 1:
        print(n)

        if n % 2 == 0:
            n //= 2
        else:
            n = 3n + 1
```

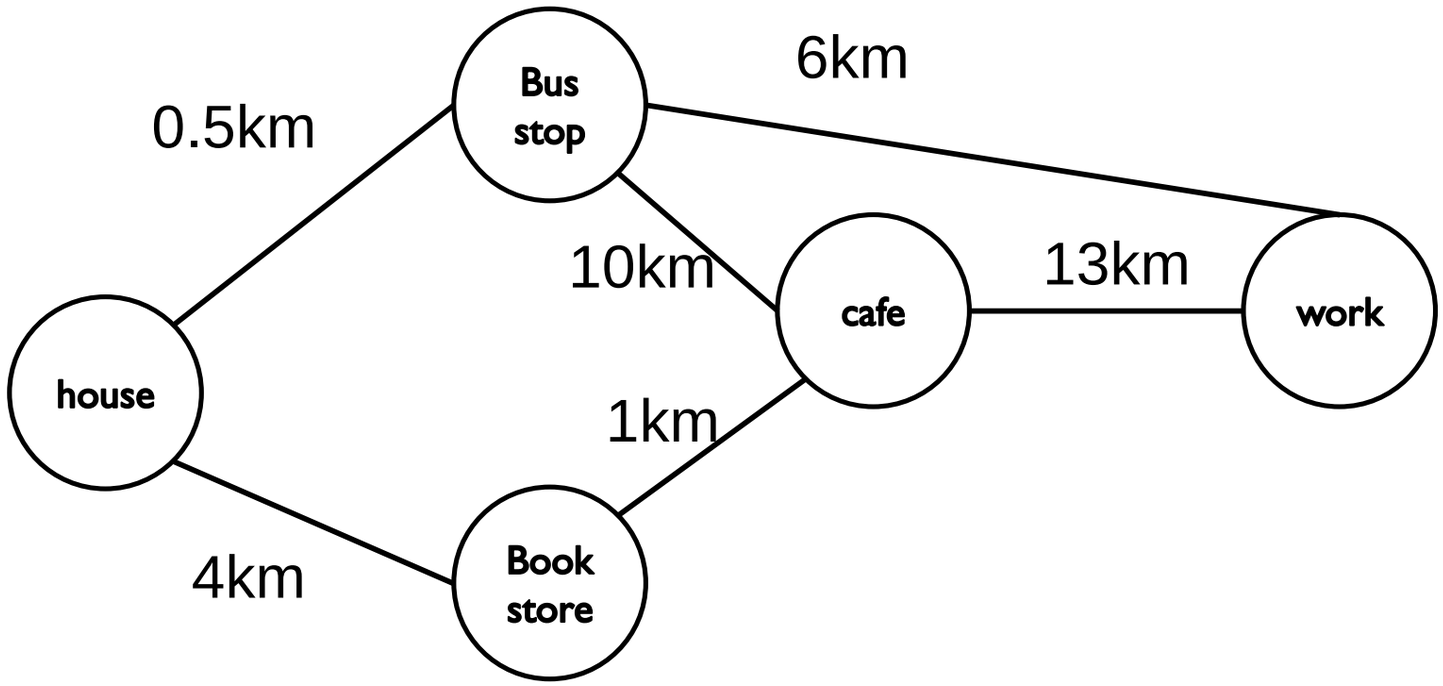
A. 4

B. 5

C. 6

D. None of the above.

What is the shortest path from ‘house’ to ‘work’?



a) House, bus stop, work
b) House, bus stop, café, work
c) House, book store, café, work
d) None of the above

Transform and conquer: Tree balancing

- What is the worst case time complexity for a binary search tree?

A. O(1)
B. O(logN)
C. O(N)
D. O(NlogN)

TSP

- The traveling salesperson problem is difficult to solve via brute force because:

A. Every vertex can only be used once
B. It is known not to have an optimal ordering
C. Computers running it are too slow
D. The number of possible orderings is O(N!)

When generating bit lists of length N, if len(partialSolution) < N, then next possible bits are always 0 and 1.

A. True

B. False

What does the following code print?

```
def update(old):
      old.append(2)
      old = [2, 3]

old = [1]
update(old)
print(old)
```

A. [1,2]
B. [2,3]
C. [1,2,2,3]
D. [1]
E. None of the above

The following code prints [2,3].

```
def clear(aList):
    while len(aList)>0:
        aList.pop()
def update(old):
    clear(old)
    for item in [2, 3]:
        old.append(item)

old = [1]
update(old)
print(old)
```

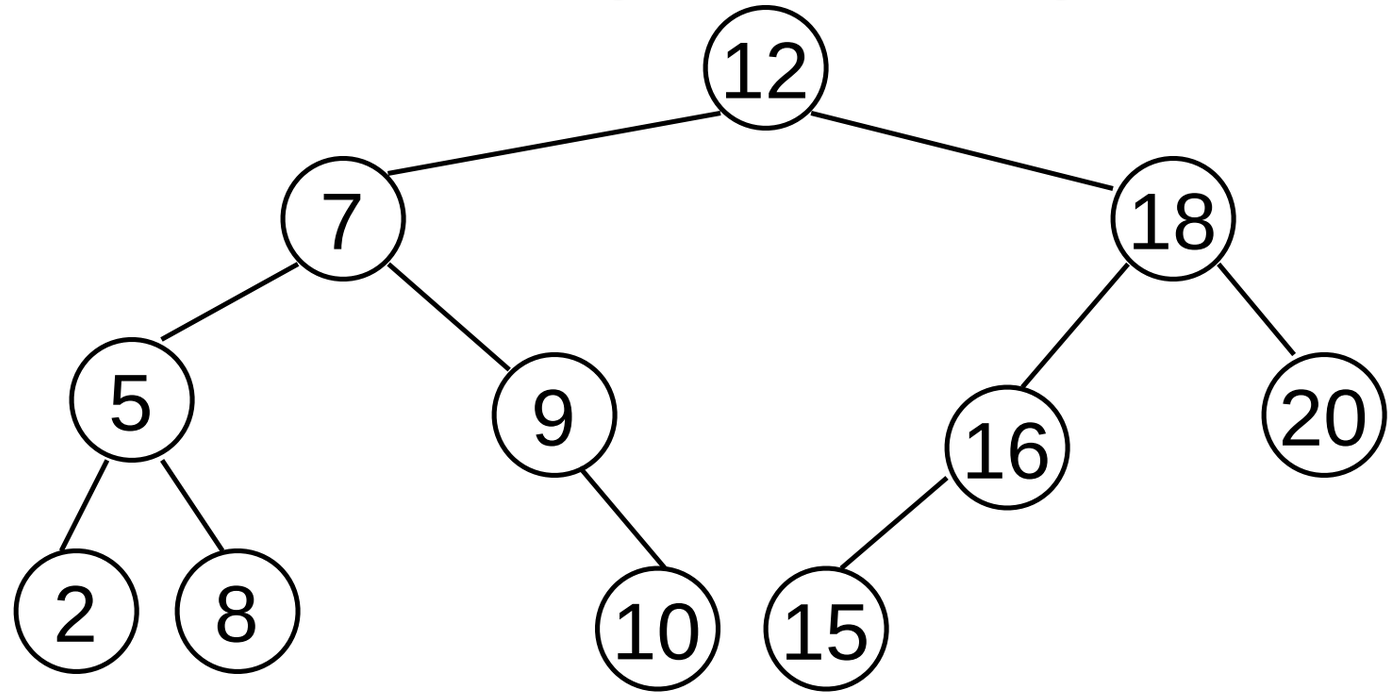
A. True

B. False

Is this a binary search tree?

Recall: In binary search tree, for every node:

- The values of each node in its left subtree is less than its value.
- The values of each node in its right subtree is greater than its value.



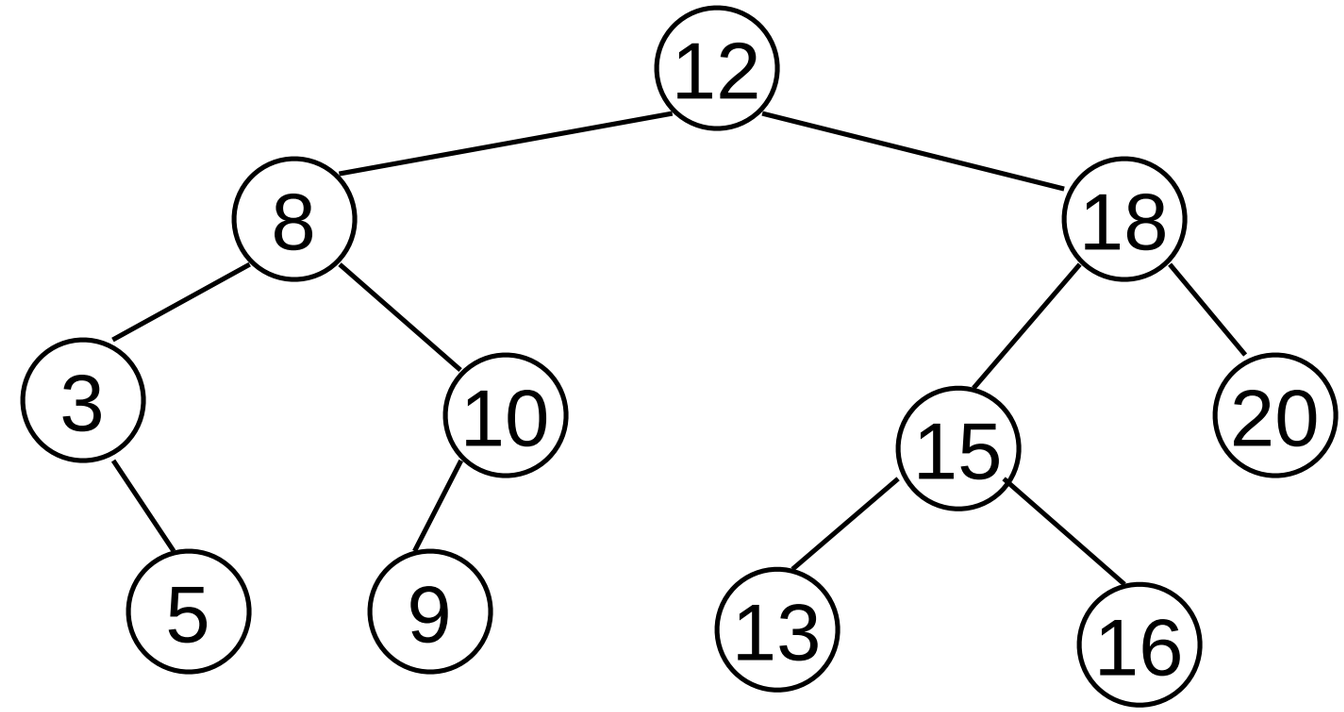
A. Yes

B. No

Is this a binary search tree?

Recall: In binary search tree, for every node:

- The values of each node in its left subtree is less than its value.
- The values of each node in its right subtree is greater than its value.

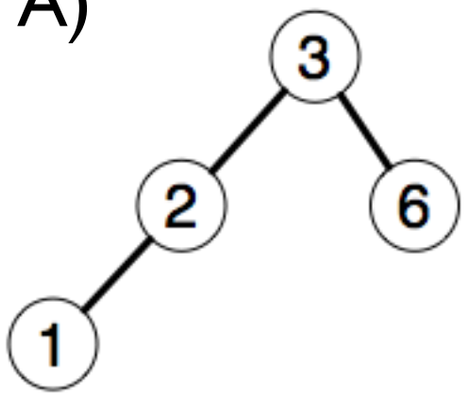


A. Yes

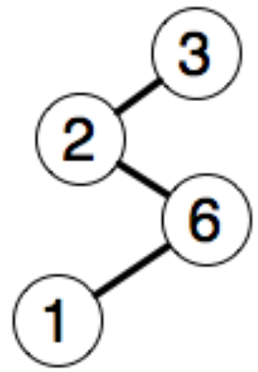
B. No

The binary tree that results from inserting the elements of the list [3,2,6,1] in this order:

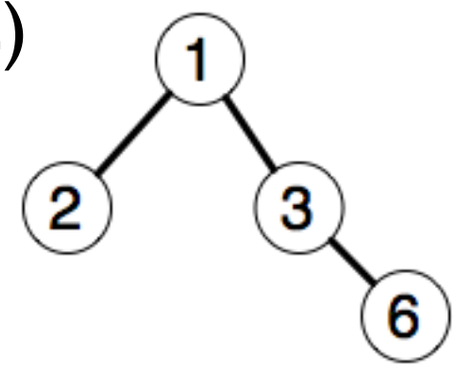
A)



B)



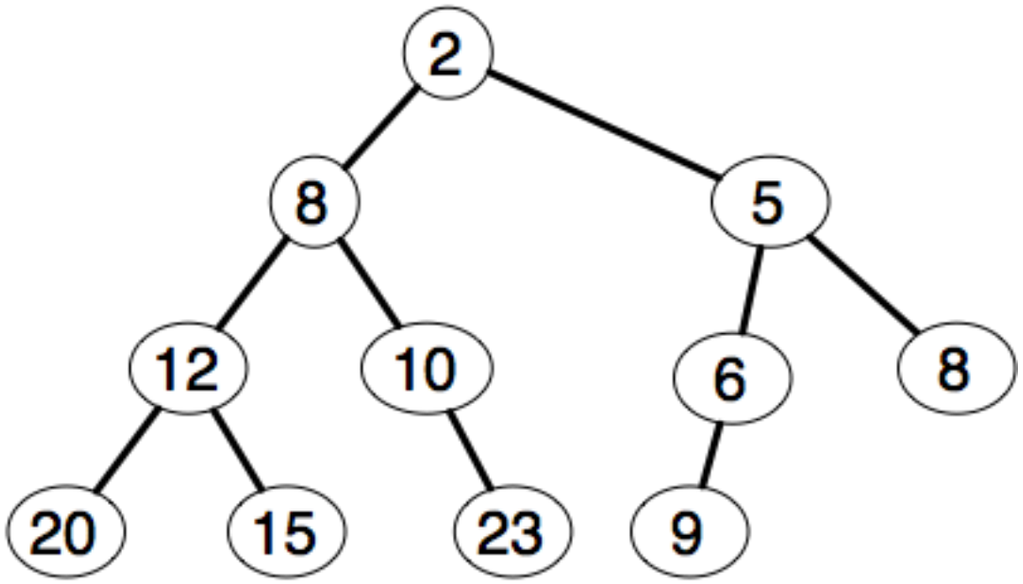
C)



D) None of the above

A.
B.
C.
D.

The following tree is a min-Heap.



A. Yes

B. No

According to the RAM model, what is the running time for power(2, 5)?

```
def power(x, N):
    'computes x to the power of N'

    value = 1
    k = 1
    while k <= N:
        value *= x
        k += 1

    return value
```

A. 5

B. 18

C. 19

D. None of the above

What is the complexity of an algorithm in Big-O notation that runs in $8N^3 + 17\ N^2 + 150$?

A. $O(8N^3)$

B. $O(N^3 + N^2)$

C. $O(N^3)$

D. $O(8N^3)$

E. $O(8N^3 + 17\ N^2 + 150)$

F. None of the above

What is the complexity of an algorithm in Big-O notation that runs in $30N \log (N^2) + 10 \log N + 8N$?

A. $O(N \log N)$

B. $O(N \log (N^2))$

C. $O(N \log (N^2) + N + \log N)$

D. None of the above

Complexity of swapElements

```
def swapElements(myList,i,j):
    temp = myList[i]
    myList[i] = myList[j]
    myList[j] = temp
```

What is the time complexity of swapElements

A. O(1)
B. O(N)
C. O(3)
D. None of the above

Note: N is the number of elements in myList

Which of the following is the worst-case for binarySearch?

```
def binarySearch(aList, target):
        low = 0
        high = len(aList)-1
        while low <= high:
              mid = (low + high) // 2
              if aList[mid] == target:
                    return mid
             if aList[mid] > target:
                    high = mid-1
              else:
                    low = mid+1
      return -1
```

A. target is not in aList
B. aList is empty
C. target is a negative value

If searching for an item stored at the root node, will you find it faster in a …

A. Balanced Tree (Height = O(log(N))
B. Unbalanced Tree (Height = O(N))
C. Equally fast on both

The best time complexity for inserting an item not already in the Binary Search Tree is?

A. O(1)
B. O(N)
C. O(log(N))
D. None of the above

The best time complexity for inserting an item into a min-Heap is?

A. O(1)
B. O(N)
C. O(log(N))
D. None of the above

The worst time complexity for inserting an item into a min-Heap is?

A. O(1)
B. O(N)
C. O(log(N))
D. None of the above

The worst time complexity for extracting an item from a min-Heap is?

A. O(1)
B. O(N)
C. O(log(N))
D. None of the above