CSCI 2270 - Zagrodzki, Ashraf, Trivedi - CS2: Data Structures

Home / My courses / Spring 2020 / CSCI2270-S20 / 16 March - 22 March / Quiz 8

Started on	Monday, 30 March 2020, 3:58 AM
State	Finished
Completed on	Monday, 30 March 2020, 4:08 AM
Time taken	9 mins 49 secs
Grade	**3.75** out of 10.00 (**38**%)

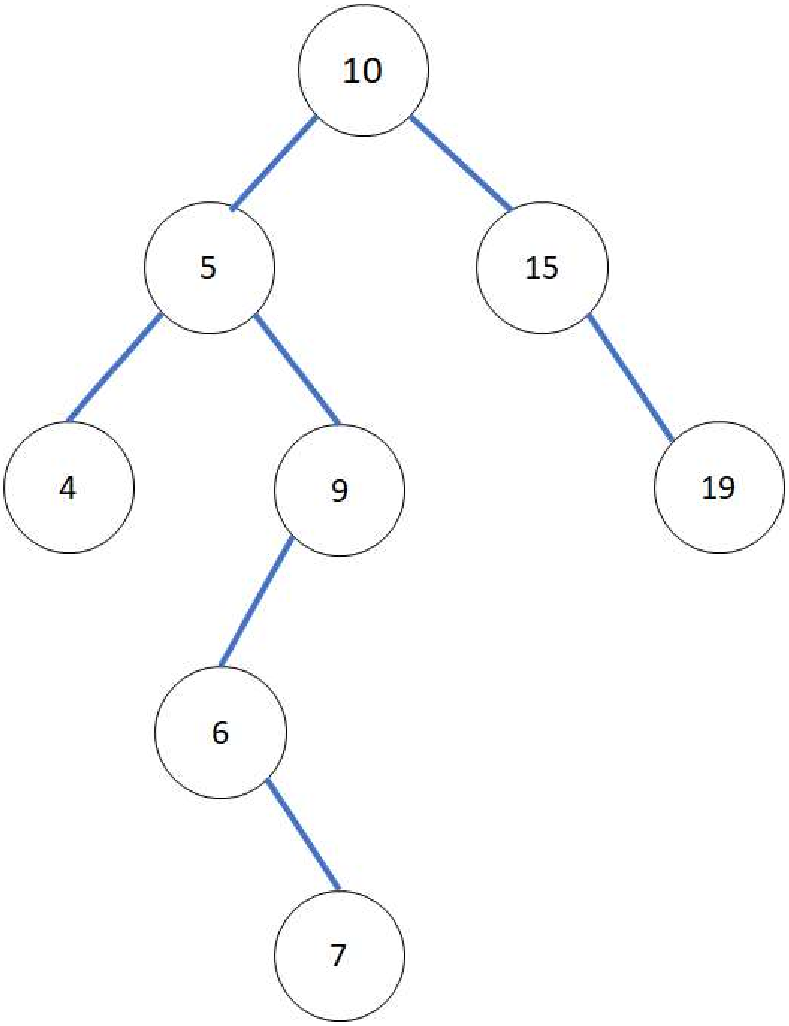
Question **1**

Correct

Mark 1.00 out of 1.00

A **balanced** binary tree is a binary tree structure in which the left and right subtrees of every node differ in height by no more than 1

Out of the following choices, which is the minimum set of nodes, if removed, will make the BST balanced?



Select one:

○ a. {19}

○ b. { }

● c. {7} ✓

○ d. {6 , 7}

○ e. {6 , 7, 19}

Your answer is correct.

The correct answer is: {7}

Question **2**

Partially correct

Mark 0.75 out of 1.00

Which of the following is true for Red-Black Trees ? Select all choices that apply!

Select one or more:

☐ a. The root of tree is always red.

☑ b. Red-Black Tree is a self-balancing Binary Search Tree (BST) ✓

☐ c. For each node in the tree, all paths from that node to any leaf nodes contain the same number of black nodes

☑ d. Every node has either red or black color. ✓

☐ e. If a node is red, then both of its children must be black, but its parent should be red.

☑ f. Every leaf (NIL) node is black ✓

Your answer is partially correct.

You have correctly selected 3.
The correct answers are: Red-Black Tree is a self-balancing Binary Search Tree (BST), Every node has either red or black color., Every leaf (NIL) node is black, For each node in the tree, all paths from that node to any leaf nodes contain the same number of black nodes

Question **3**

Incorrect

Mark 0.00 out of 1.00

It is possible to have a red-black tree with no red nodes.

Select one:

○ True

● False ✗

The correct answer is 'True'.

Question **4**

Incorrect

Mark 0.00 out of 1.00

After performing a left rotation on the root node of the following tree, select the correct preorder traversal from the given options.

```
                   10
                  /  \
                 8   15
                     / \
                   14  19
```

Select one:

○ 8,14,10,19,15

○ 19,15,14,10,8

○ 15,10,8,14,19

◉ None of these ✖

○ 8,10,14,15,19

Your answer is incorrect.

The correct answer is: 15,10,8,14,19

Question **5**

Correct

Mark 1.00 out of 1.00

The following descriptions are based on C++ Standard Library queue and vector.

We use push() ✓ to Inserts a new element at the end of the queue, pop() ✓ function to remove an element from the front of the queue and size() ✓ to check the size of the queue.

We use push_back() ✓ to push the elements into a vector from the back, pop_back() ✓ function to pop or remove elements from a vector from the back and size() ✓ to check the size of the vector

push() pop_back() size() insert() assign() pop() push_back() empty()

Your answer is correct.

The correct answer is:

The following descriptions are based on C++ Standard Library queue and vector.

We use [push()] to Inserts a new element at the end of the queue, [pop()] function to remove an element from the front of the queue and [size()] to check the size of the queue.

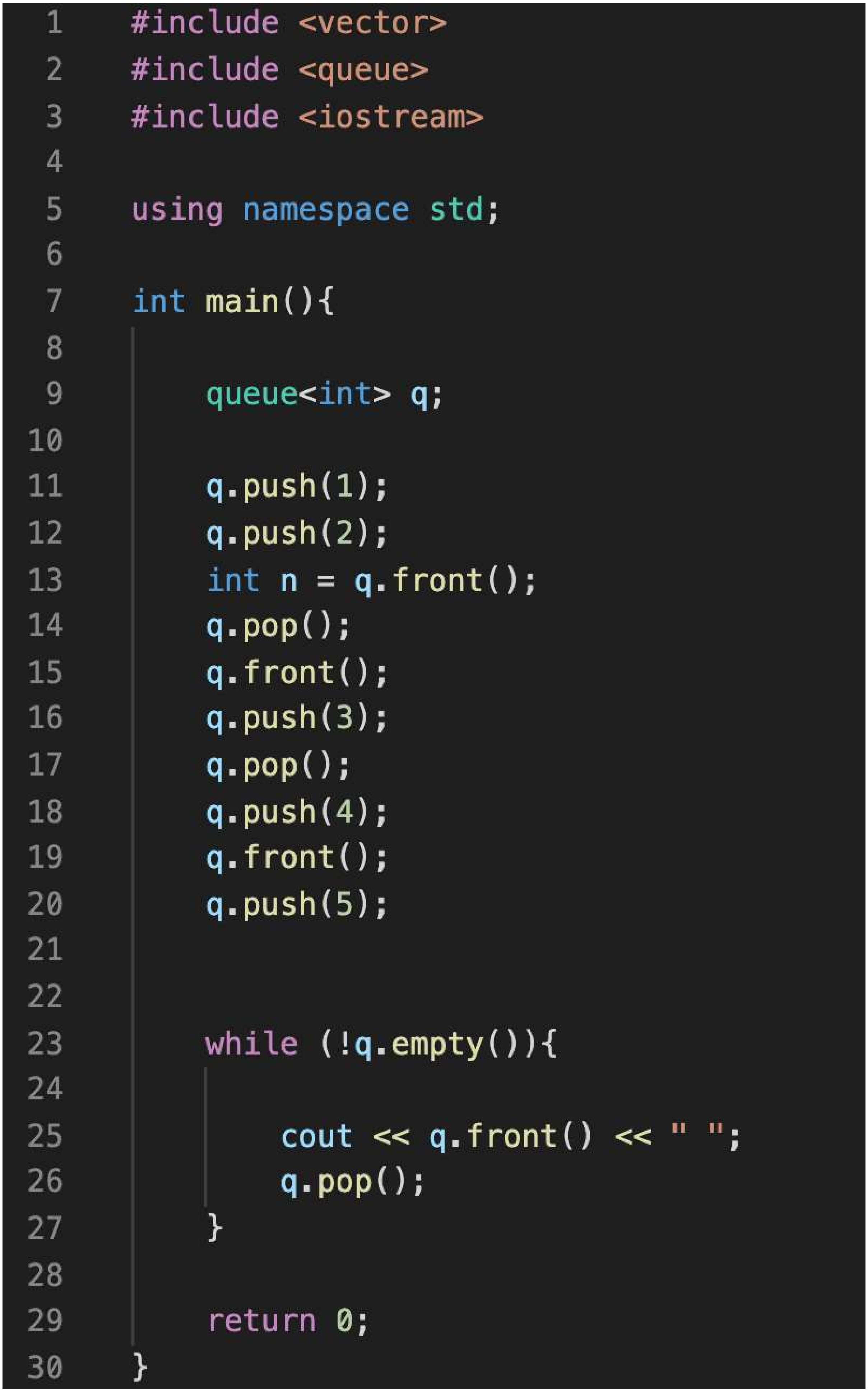
We use [push_back()] to push the elements into a vector from the back, [pop_back()] function to pop or remove elements from a vector from the back and [size()] to check the size of the vector

Question **6**

Incorrect

Mark 0.00 out of 1.00

What is the output of the following code segment?



```cpp
#include <vector>
#include <queue>
#include <iostream>

using namespace std;

int main(){

    queue<int> q;

    q.push(1);
    q.push(2);
    int n = q.front();
    q.pop();
    q.front();
    q.push(3);
    q.pop();
    q.push(4);
    q.front();
    q.push(5);


    while (!q.empty()){

        cout << q.front() << " ";
        q.pop();
    }

    return 0;
}
```

Select one:

○ 3 4 5

○ 5 4 1

● 1 4 5 ✗

○ 5

Your answer is incorrect.

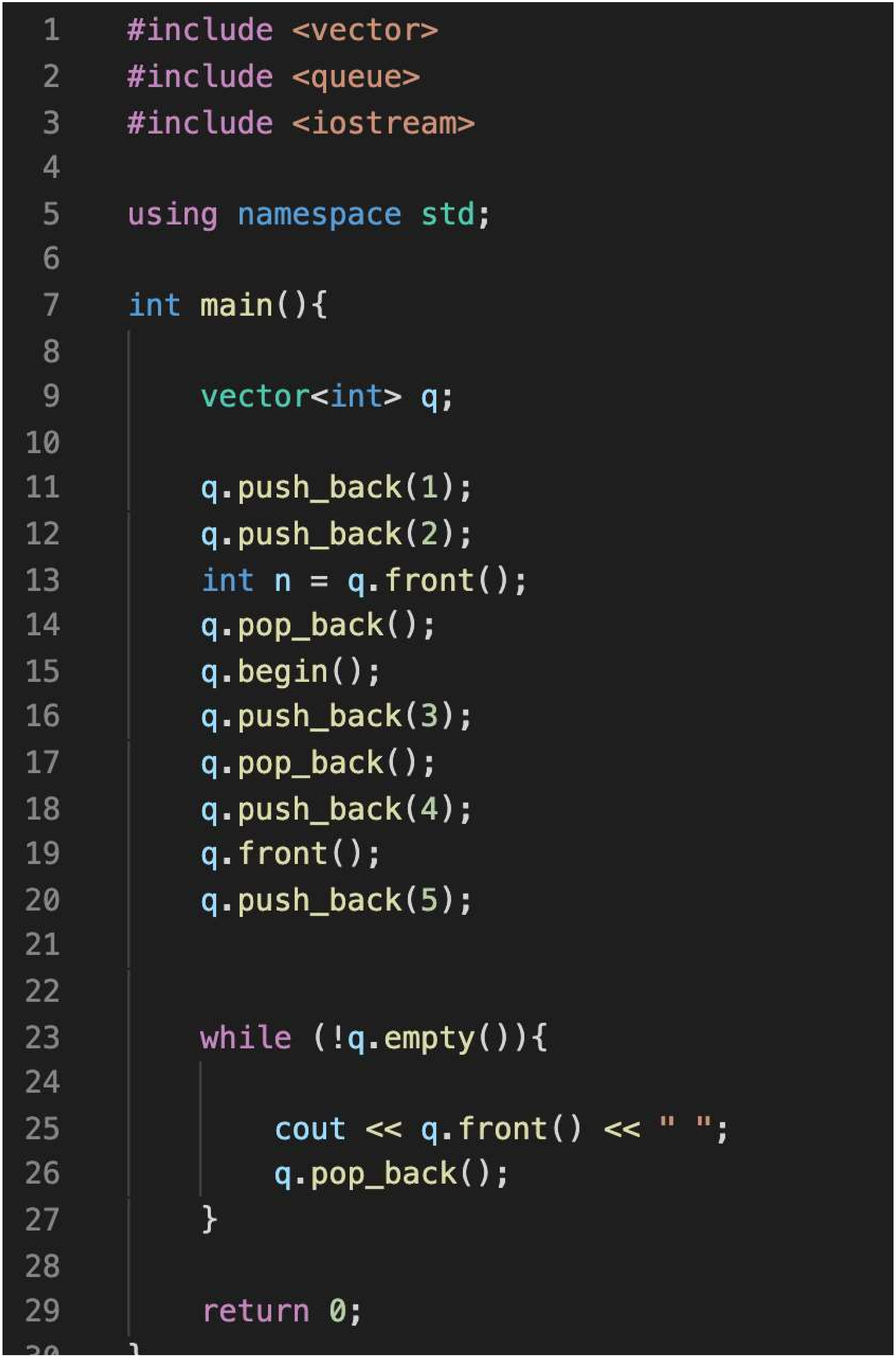
The correct answer is: 3 4 5

Question **7**

Incorrect

Mark 0.00 out of 1.00

What is the output of the following code segment?



```
#include <vector>
#include <queue>
#include <iostream>

using namespace std;

int main(){

    vector<int> q;

    q.push_back(1);
    q.push_back(2);
    int n = q.front();
    q.pop_back();
    q.begin();
    q.push_back(3);
    q.pop_back();
    q.push_back(4);
    q.front();
    q.push_back(5);


    while (!q.empty()){

        cout << q.front() << " ";
        q.pop_back();
    }

    return 0;
}
```

```
        }
```

Select one:

● 5 4 1 ✖

○ 3 4 5

○ 1 1 1

○ 5

Your answer is incorrect.

The correct answer is: 1 1 1

Question **8**

Incorrect

Mark 0.00 out of 1.00

In a depth-first traversal (DFS), one can use a [Tree] ✖ for backtracking.

In a breadth-first traversal (BFS), one can use a [Linked List] ✖ to traverse the graph level by level.

[Array] [Linked List] [Stack] [Queue] [Tree]

Your answer is incorrect.

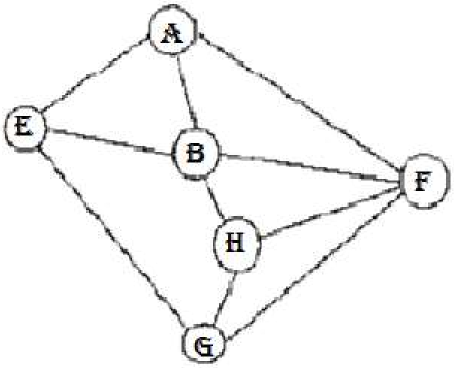
The correct answer is:
In a depth-first traversal (DFS), one can use a [Stack] for backtracking.

In a breadth-first traversal (BFS), one can use a [Queue] to traverse the graph level by level.

Question **9**

Correct

Mark 1.00 out of 1.00



Consider the above graph.

Which of the following answers shows a valid DFS and BFS on the graph?

Select one:

○ a. **DFS**: B H A G F E ; **BFS**: H B A E G F

○ b. **DFS**: G H F B A E; **BFS**: H G E A B F

○ c. **DFS**: E A G F B H; **BFS**: H B A F G E

● d. **DFS**: G E A F B H; **BFS**: H B F G A E ✓

Your answer is correct.

The correct answer is: **DFS**: G E A F B H; **BFS**: H B F G A E

Question **10**

Incorrect

Mark 0.00 out of 1.00

Using an adjacency list to represent a graph always uses less space than using an adjacency matrix.

Select one:

● True ✗

○ False

The correct answer is 'False'.