# Midterm 2: Practice multiple-choice questions

#### **Question 1**

The following class declaration would be used to create which data structure?

```
class Node {
  int data;
  vector<Node*> adj;
}
```

- a. Binary Search Tree
- b. Hash Table
- c. Heap
- d. Graph

#### **Question 2**

Which of these data structures would be the most useful when storing a social media used by billions of people, where the main feature is a search bar that lets users look up other users by entering their name?

- a. Binary Search Tree
- b. Hash Table
  - c. Heap
  - d. Graph

## **Question 3**

#### What will be printed when the following code is run?

```
#include <iostream>
   struct Node {
     int key;
     Node* left;
     Node* right;
     Node(int k) {
       this->key = k;
       this->left = this->right = NULL;
     }
   };
   void print(Node* n) {
     if (n == NULL) return;
     std::cout << n->key << " ";
     print(n->left);
     print(n->right);
   int main() {
     Node* root = new Node(12);
     root->left = new Node(6);
     root->right = new Node(15);
     root->left->left = new Node(2);
     root->left->right = new Node(8);
     print(root);
   }
 a. 2861512
 b. 15 12 8 6 2
c. 12 6 2 8 15
 d. 2681215
```

## **Question 4**

Which of the following algorithms could be used to find the shortest path between two points on a grid?

- a. Upward heap repair
- b. Breadth-first search
- c. Depth-first search
- d. Recursive traversal

# **Question 5**

Given a hash table with 4 boxes that uses a suitably random hash function, what is the largest number of elements that you can insert while being *certain* that there is no collision in the table?

- a. 1
- b. 2
- c. 3
- d. 4