

Name: _____

Date: _____

The following items should be typed and printed. Your results should also be submitted to the course website.

1. Assume that we are utilizing a doubly linked list of unsorted data. What would be the running time of the data structure for the following functions?

- SEARCH
- INSERT
- DELETE
- MINIMUM
- MAXIMUM
- SUCCESSOR
- PREDECESSOR

2. Implement a data structure that will allow you to implement two stacks in a single array. Be sure to explain how you will accomplish this in a paragraph or less and then provide the code to support this implementation.

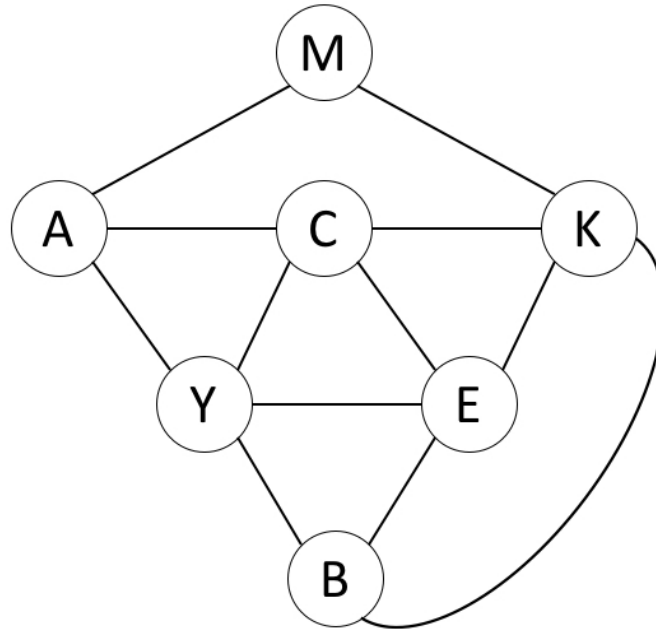
3. Draw a binary search tree by inserting the following numbers:

{8, 1, 5, 4, 9, 10, 6, 3, 15}

4. From the tree in the previous question, determine the successor and predecessor of 1.

5. Determine the maximum number of nodes from a tree with a height of 7. What would be the minimum number of nodes for this same tree? Show your work.

6. Represent the following undirected graph as both an adjacency list and adjacency matrix.



7. Assume that the previous graph was directed. How many edges will you have in the graph?

8. Write an algorithm that will find the third smallest number in a binary search tree.

9. Draw a tree with $n = 9$ nodes. Demonstrate that there are $n + 1$ null references that represent child nodes.

10. What are the advantages and disadvantages of the various hash table methods for managing collisions? State each of them and explain.

Practical Examination

Be sure to save any code as requested in a directory named exam (/home/ua123/exam) on the code server where ua123 is your username. Be sure to upload your results to the course website as well. Any issues with paths, filenames, etc. will result in a deduction of points. All data structures must be custom and implemented by you (do not use standard libraries for this task).

1. Create a class named `UExam2Problem1`. Prompt the user to enter integers from the command line until the word `END` is typed. Each of the numbers entered should be added to a doubly linked list of type `UANode`. Print out each of the items in this linked list to the console.

Next, create a **new** doubly linked list that has no references to the objects of the previous linked list. This new linked list should be the reversal of the previous linked list. Add 10 to each of the keys in the linked list. Print out the contents of this new linked list to the console and print out the results of the old linked list to the console.

The doubly linked list node class should be defined by you and named `UANode`. It contains a single integer field named `key`. You should override the `toString()` method to output the `Object Value: key, Previous Value: prev.key, Next Value: next.key` where `key` is the actual value stored in the variable and `prev/next` both serve as references to the previous and next elements in the linked list. If `prev` or `next` are null, output `NULL` to the console.

Sample Input:

```
10
20
30
END
```

Sample Output:

```
Object Value: 10,      Next Value: 20,      Previous Value: NULL
Object Value: 20,      Next Value: 30,      Previous Value: 10
Object Value: 30,      Next Value: NULL,    Previous Value: 20
```

*** Reversing ***

```
Object Value: 40,      Next Value: 30,      Previous Value: NULL
Object Value: 30,      Next Value: 20,      Previous Value: 40
Object Value: 20,      Next Value: NULL,    Previous Value: 30
```

*** Original List ***

```
Object Value: 10,      Next Value: 20,      Previous Value: NULL
Object Value: 20,      Next Value: 30,      Previous Value: 10
Object Value: 30,      Next Value: NULL,    Previous Value: 20
```