Final

Instructions

The exam window is from **Friday, June 11 at 8:00 AM** to **Saturday, June 12 at 7:59 AM** (all times are in San Diego time). From the moment you begin the exam, you will have **180 minutes** to complete it. However, Saturday, June 12 at 7:59 AM San Diego time is a hard deadline, so be sure to start as early as possible.

No collaboration whatsoever during the exam! The exam is open book, open internet, and open notes, but **no Chegg or any similar services!** If you are caught cheating on the exam, you will receive an **F in the course**.

For each of the exam problems, you have to click the "Submit" button to submit your response, and you have **unlimited attempts** without penalty. However, **you will not know if your answer is correct!** Ed will tell you whether or not you have submitted, but we will grade the correctness of your responses after the exam. Each of the **80 total problems** on the exam is worth exactly 1 point. There is no global "Submit" button: as long as you've clicked the "Submit" button for each individual question, your answers are submitted.

For the sake of fairness for all students, including those who take the exam earlier in the window, **you will not be able to ask any questions!** Instead, after you finish the exam, you can mention any questions that you felt were ambiguous via the following Google Form:

https://forms.gle/ZZ1c2eiYNaMRPepi9

Then, *after* the exam, we will review the responses to this question to determine if we need to adjust the grading in any way. Please open the form now so you don't lose access to it if you run out of time on the exam.

Week 1: C++ Review

This section has 5 questions (below). You may need to scroll down to see them.

Report ambiguous questions: https://forms.gle/ZZ1c2eiYNaMRPepi9

Question 1 *Submitted Jun 1st 2021 at 1:15:33 pm*

Problem 1.1: What will be printed by this program?

```
int main() {
    int x = 3;
    int y = 7;
    int* z = &y;

    (*z) = (*z) + y;
    y = y + x;

    cout << y;
}
```


17

Question 2 *Submitted Jun 1st 2021 at 1:15:37 pm*

Problem 1.2: True/False: The code snippet in Problem 1.1 will result in a memory leak.

- ○ True
- ◉ False

Question 3 *Submitted Jun 1st 2021 at 1:15:41 pm*

Problem 1.3: In the code snippet in Problem 1.1, which of the following will be stored in the Heap? Select all that apply.

- ☐ `x`
- ☐ `y`

- [ ] `z`
- [ ] The object `z` points to
- [x] None of the above

Question 4 *Submitted Jun 1st 2021 at 1:15:48 pm*

Problem 1.4: What is the tightest worst-case Big-O time complexity of the following code snippet?

```
for(int i = 1; i < n; i *= 2) {
    cout << "i: " << i << endl;

    for(int j = 1; j < n/2; ++j) {
        cout << "i + j: " << (i+j) << endl;
    }

    for(int k = 1; k < n; k *= 2) {
        cout << "i * k: " << (i*k) << endl;
    }
}
```


- [ ] O(log *n*)
- [ ] O(*n*)
- [x] O(*n* log *n*)
- [ ] O(n^2)
- [ ] None of the above

Question 5 *Submitted Jun 1st 2021 at 1:15:58 pm*

Problem 1.5: What is the tightest worst-case Big-O time complexity of the following code snippet, which utilizes the C++ STL's `list` class (which is implemented as a Doubly Linked List) and `max_element` function (which iterates over the elements in the given range and returns the maximum value)?

```
int main() {
```

```
    list<int> my_list;
    for(int i = 1; i <= n/2; ++i) {
        my_list.push_back(i);
    }
    for(int i = n/2 + 1; i <= n; ++i) {
        my_list.push_front(i);
    }
    while(!my_list.empty()) {
        auto curr = max_element(my_list.begin(), my_list.end());
        cout << *curr << endl;
        my_list.erase(curr);
    }
    return 0;
}
```

- O(log n)
- O(n)
- O(n log n)
- O(n^2)
- O(n^2 log n)
- O(n^3)
- None of the above

Week 2: Trees and BSTs

This section has 7 questions (below). You may need to scroll down to see them.

Report ambiguous questions: https://forms.gle/ZZ1c2eiYNaMRPepi9

Question 1 *Submitted Jun 7th 2021 at 1:41:56 pm*

Problem 2.1: You are given a `BinaryTree` class consisting of `Node` objects that are defined as follows:

```
class Node {
    public:
        Node* left_child;
        Node* right_child;
        char symbol;
};
```


You are writing a recursive function to print the symbols of all of the `Node` objects via a **post-order** traversal in which you recurse down the left child before recursing down the right child:


```
void print_postorder(Node* &curr) {
    // YOUR CODE HERE
}
```


Rearrange the following lines of code such that they would implement the function correctly.

Note: You will need to use all of the lines of code.

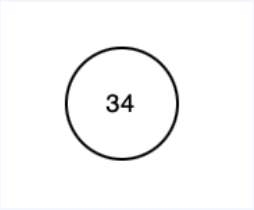
Drag from here

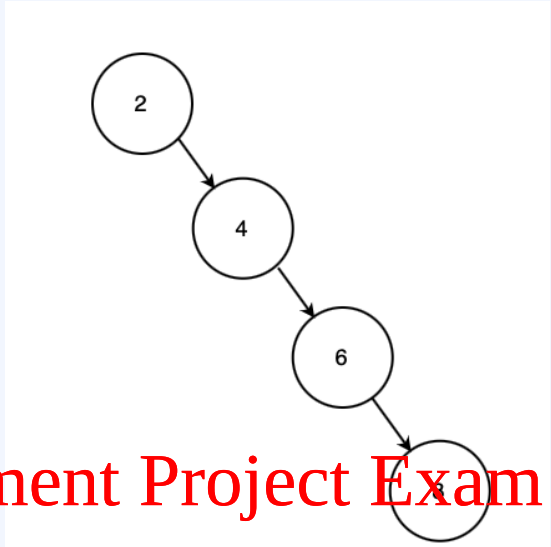
Drop blocks here

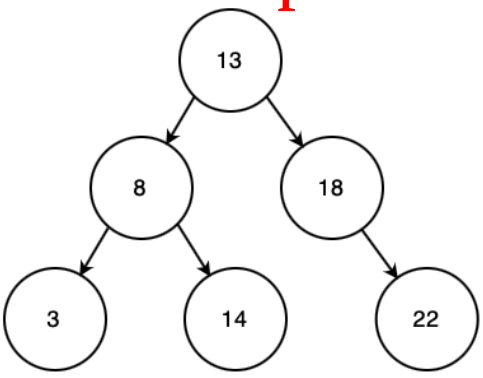
```
if(curr == nullptr) {
return;
}
print_postorder(curr->left_child);
print_postorder(curr->right_child);
cout << curr->symbol << endl;
```

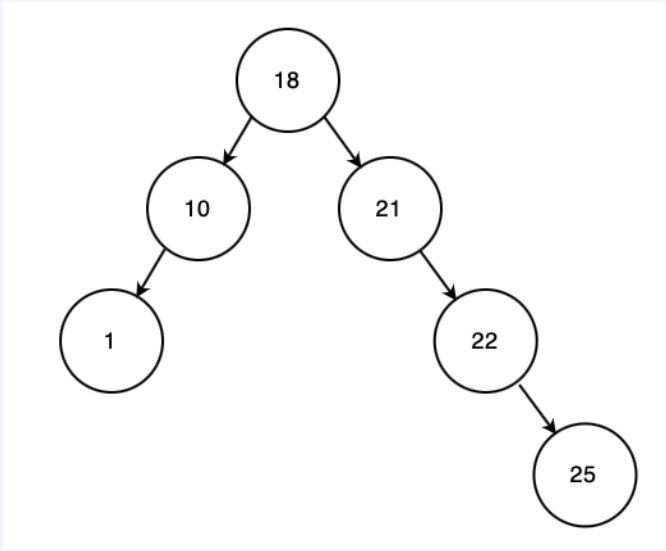
Question 2 *Submitted Jun 1st 2021 at 1:31:36 pm*

Problem 2.2: Which of the following are valid Binary Search Trees? Select all that apply.

- [x] 

- [x] 

- [ ] 



None of the above

Question 3 *Submitted Jun 1st 2021 at 1:31:40 pm*

Problem 2.3: True/False: The right-most node in a Binary Search Tree does not have a successor.

Note: The "right-most node" is the node you reach by starting at the root and traversing down right-child pointers as far as possible.

True

False

Question 4 *Submitted Jun 1st 2021 at 1:31:44 pm*

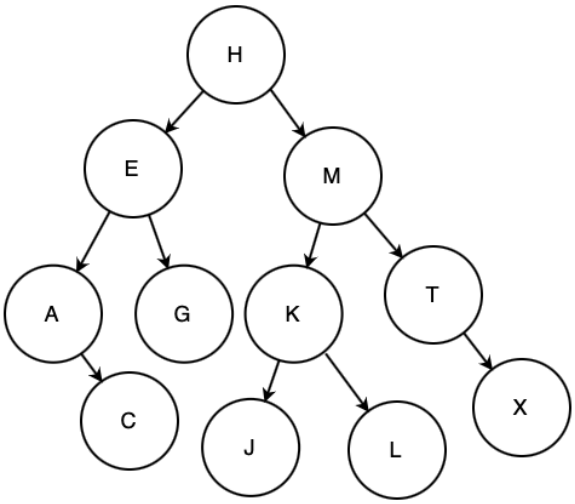
Problem 2.4: True/False: In a Binary Search Tree, if node *x* and node *y* are both children of node *z*, it is impossible for *y* to be the successor of *x*.

True

False

Question 5 *Submitted Jun 1st 2021 at 1:33:20 pm*

Problem 2.5: You are given the following Binary Search Tree:



You attempt to find the successor of node **H** using the successor algorithm discussed in this course. Which nodes will you visit, and in what order?

Note: Include both H and its successor in your answer.

Note: Order them such that the top item in your answer is the first node to be visited and the bottom item in your answer is the last node to be visited.

Note: You will not necessarily include all elements in your answer.

Drag from here

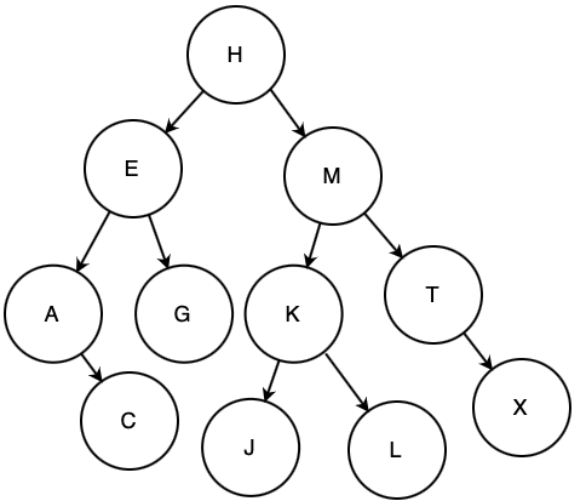
- T
- X
- G
- L
- E
- A
- C

Drop blocks here

- H
- M
- K
- J

Question 6 *Submitted Jun 5th 2021 at 7:18:51 am*

Problem 2.6: You are given the following Binary Search Tree:



You attempt to find the successor of node **L** using the successor algorithm discussed in this course. Which nodes will you visit, and in what order?

Note: Include both L and its successor in your answer.

Note: Order them such that the top item in your answer is the first node to be visited and the bottom item in your answer is the last node to be visited.

Note: You will not necessarily include all elements in your answer.

Drag from here

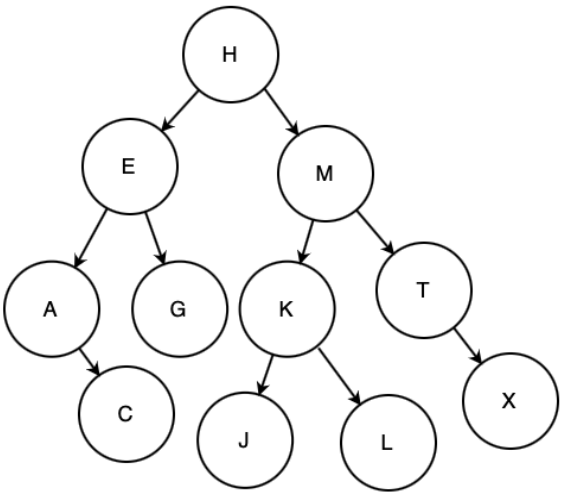
- H
- A
- G
- X
- C
- J
- E
- T

Drop blocks here

- L
- K
- M

Question 7 *Submitted Jun 5th 2021 at 7:33:55 am*

Problem 2.7: You are given the following Binary Search Tree:



After removing node **M** using the remove algorithm discussed in this course, what is the output of a level-order traversal of the resulting tree?

Note: Order them such that the top item in your answer is the first node to be visited and the bottom item in your answer is the last node to be visited.

H

E

T

A

G

K

X

C

J

L

Week 3: Self-Balancing BSTs

This section has 10 questions (below). You may need to scroll down to see them.

Report ambiguous questions: https://forms.gle/ZZ1c2eiYNaMRPepi9

Question 1 *Submitted Jun 5th 2021 at 10:15:41 am*

Problem 3.1: True/False: In practice, we expect the height of a Red-Black Tree to be less than the height of an AVL Tree containing the same elements.

- ○ True
- ◉ False

Question 2 *Submitted Jun 5th 2021 at 10:15:47 am*

Problem 3.2: True/False: In practice, we expect the height of a Red-Black Tree to be less than the height of a Randomized Search Tree containing the same elements.

- ◉ True
- ○ False

Question 3 *Submitted Jun 5th 2021 at 10:16:01 am*

Problem 3.3: True/False: The tightest worst-case Big-O time complexity to insert an element into a regular (non-self-balancing) Binary Search Tree is larger than the tightest worst-case Big-O time complexity to insert an element into a Randomized Search Tree.

Example: O(n) would be "larger" than O(1).

- ○ True
- ◉ False

Question 4 *Submitted Jun 5th 2021 at 10:16:24 am*

Problem 3.4: What is the minimum possible number of black nodes in a Red-Black Tree with 7 nodes?

Note: Don't worry about whether or not there exists an insertion order to produce such a tree. Assume you can design the tree as you wish, as long as it's a valid Red-Black Tree.

3

Question 5 *Submitted Jun 5th 2021 at 10:16:28 am*

Problem 3.5: What is the maximum possible number of black nodes in a Red-Black Tree with 9 nodes?

Note: Don't worry about whether or not there exists an insertion order to produce such a tree. Assume you can design the tree as you wish, as long as it's a valid Red-Black Tree.

7

Question 6 *Submitted Jun 5th 2021 at 10:16:37 am*

Problem 3.6: What is the maximum possible height of an AVL Tree with 8 nodes?

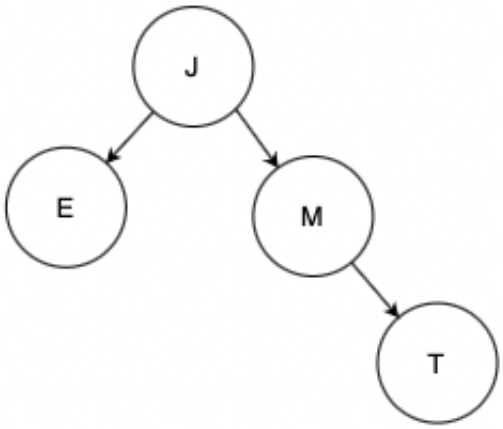
Note: Define the height of a tree to be the number of edges in the longest path from the root to a leaf.

Note: Don't worry about whether or not there exists an insertion order to produce such a tree. Assume you can design the tree as you wish, as long as it's a valid AVL Tree.

3

Question 7 *Submitted Jun 5th 2021 at 10:17:17 am*

Problem 3.7: You are given the following AVL Tree:



After inserting **P**, what is the output of a level-order traversal of the resulting tree?

Note: Order them such that the top item in your answer is the first node to be visited and the bottom item in your answer is the last node to be visited.

J

E

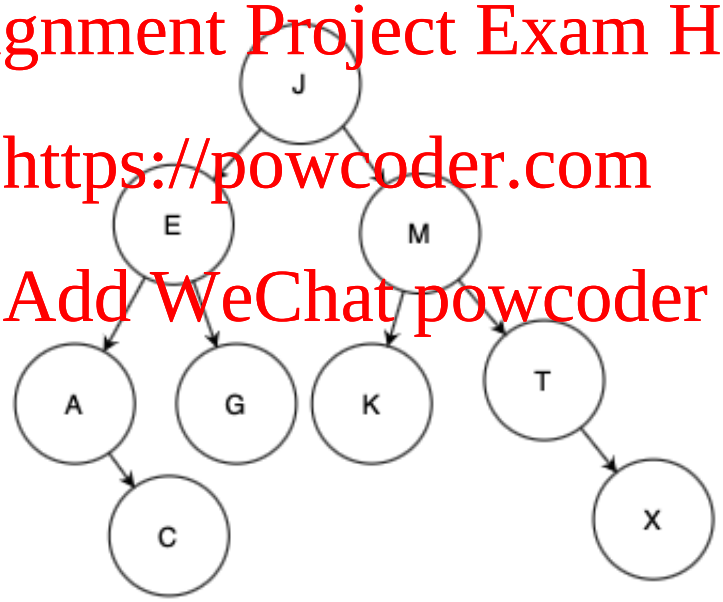
P

M

T

Question 8 *Submitted Jun 5th 2021 at 10:19:00 am*

Problem 3.8: You are given the following AVL Tree:



After inserting **D**, what is the output of a level-order traversal of the resulting tree?

Note: Order them such that the top item in your answer is the first node to be visited and the bottom item in your answer is the last node to be visited.

J

E

M

C

G

K

T

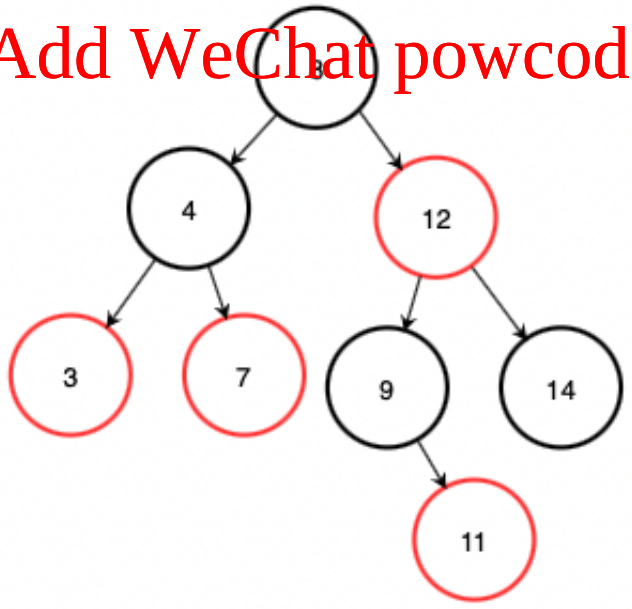
A

D

X

Question 9 *Submitted Jun 5th 2021 at 10:20:34 am*

Problem 3.9: You are given the following Red-Black Tree:



After inserting **2**, what is the output of a level-order traversal of the resulting tree?

Note: Order them such that the top item in your answer is the first node to be visited and the bottom item in your answer is the last node to be visited.

8

4

12

3

7

9

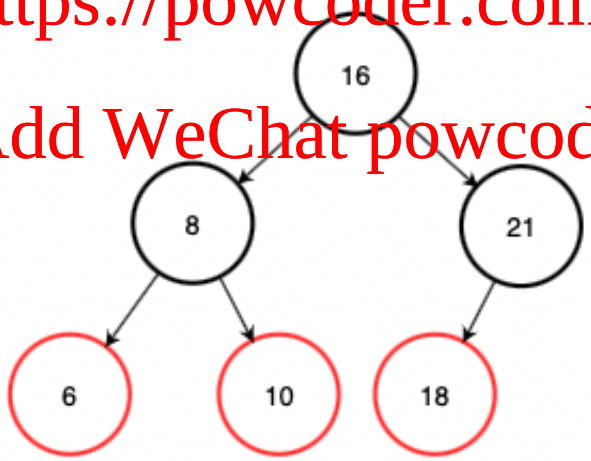
14

2

11

Question 10 *Submitted Jun 5th 2021 at 10:10:33 pm*

Problem 3.10: You are given the following Red-Black Tree:



After inserting **4**, how many nodes will be recolored during the insertion algorithm?

3

Week 4: Lexicons

This section has 10 questions (below). You may need to scroll down to see them.

Report ambiguous questions: https://forms.gle/ZZ1c2eiYNaMRPepi9

Question 1 *Submitted Jun 5th 2021 at 11:11:05 am*

Problem 4.1: True/False: After inserting *n* unique words, each of length *k* and each beginning with a different letter, into both a Multiway Trie and a Ternary Search Tree (both initially empty), the number of nodes in the resulting Multiway Trie is larger than the number of nodes in the resulting Ternary Search Tree.

Note: Unique words can share letters, but the overall words must be different.

- (●) True
- ( ) False

Question 2 *Submitted Jun 5th 2021 at 11:11:11 am*

Problem 4.2: True/False: After inserting words into a Multiway Trie that was initially empty, assuming you never remove any words, all word nodes in the Multiway Trie will be leaves.

- ( ) True
- (●) False

Question 3 *Submitted Jun 5th 2021 at 11:11:27 am*

Problem 4.3: You insert the following words, in this exact order, into a Multiway Trie that is initially empty:

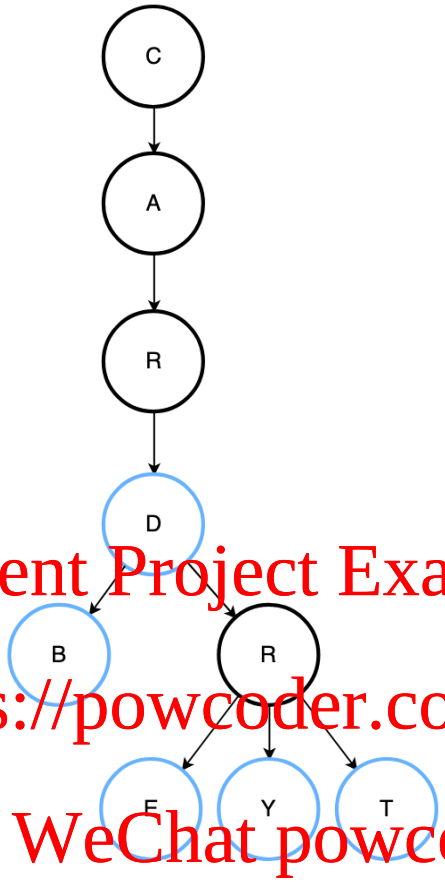
```
CLASSES
COMMUTER
COMPANY
COMPUTER
LAST
```

How many nodes are in the resulting Multiway Trie?

27

Question 4 *Submitted Jun 7th 2021 at 1:45:50 pm*

Problem 4.4: You are given the following Ternary Search Tree:



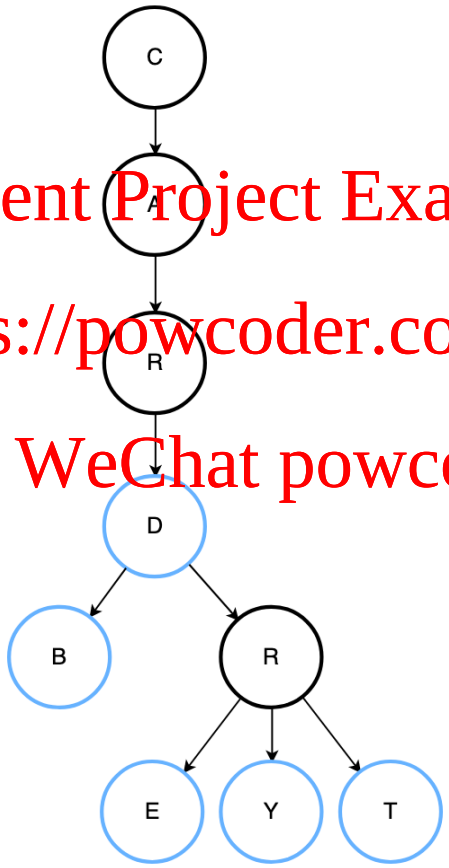
Which of the following words exist in the Ternary Search Tree? Select all that apply.

- [ ] CAR
- [x] CARB
- [ ] CARBD
- [x] CARD
- [ ] CARDB
- [ ] CARDRY

- [x] CARE
- [x] CARRY
- [ ] CARRT
- [x] CART

Question 5 *Submitted Jun 5th 2021 at 11:11:54 am*

Problem 4.5: You are given the following Ternary Search Tree:

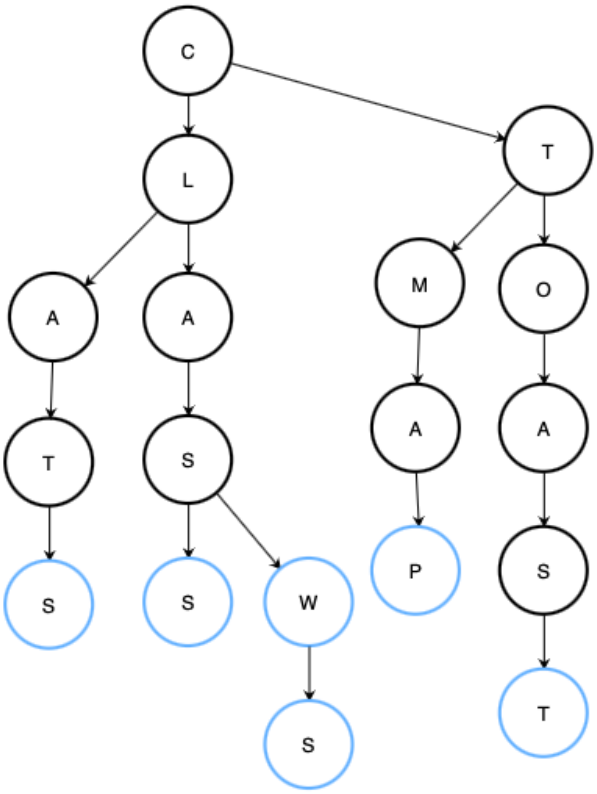


How many new nodes would be created if you were to insert the word `CHART`?

4

Question 6 *Submitted Jun 5th 2021 at 11:12:19 am*

Problem 4.6: You are given the following Ternary Search Tree:



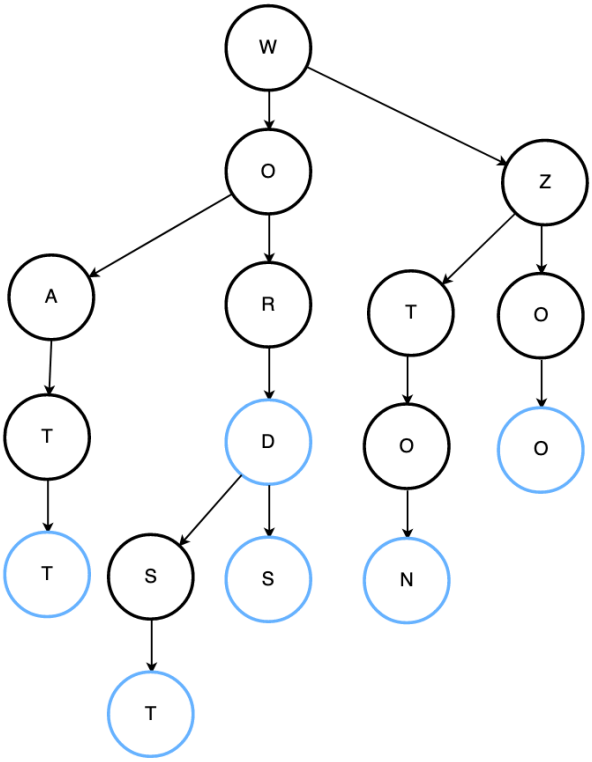
Assuming no words have ever been removed (i.e., the Ternary Search Tree was created only using insert operations), which of the following statements are true? Select all that apply.

- [x] `TOAST` must have been inserted before `MAP`
- [ ] `CLAW` must have been inserted before `CLAWS`

- [ ] `CATS` must have been inserted before `CLASS`
- [x] `CLASS` must have been inserted before `MAP`
- [ ] None of the above

Question 7 *Submitted Jun 5th 2021 at 11:12:35 am*

Problem 4.7: You are given the following Ternary Search Tree:



Assuming no words have ever been removed (i.e., the Ternary Search Tree was created only using insert operations), which of the following statements are true? Select all that apply.

- [ ] `WORD` must have been inserted before `WATT`

- [ ] `WORDS` must have been inserted before `WORST`

- [ ] `WORD` must have been inserted before `TON`
- [x] `ZOO` must have been inserted before `TON`
- [ ] None of the above

Question 8 *Submitted Jun 5th 2021 at 11:12:46 am*

Problem 4.8: What is the minimum possible number of nodes in a Multiway Trie containing 7 unique words of length 5?

Note: Unique words can share letters, but the overall words must be different.

12

Question 9 *Submitted Jun 5th 2021 at 11:12:48 am*

Problem 4.9: Assuming no remove operations have ever been performed, what is the maximum possible height of a Multiway Trie containing 7 words of length 5?

Note: Define the height of a tree to be the number of edges in the longest path from the root to a leaf.

5

Question 10 *Submitted Jun 5th 2021 at 11:12:51 am*

Problem 4.10: Assuming no remove operations have ever been performed, what is the maximum possible height of a Ternary Search Tree containing 7 words of length 5?

Note: Define the height of a tree to be the number of edges in the longest path from the root to a leaf.

10

Week 5: Hashing Data Structures

This section has 10 questions (below). You may need to scroll down to see them.

Report ambiguous questions: https://forms.gle/ZZ1c2eiYNaMRPepi9

Question 1 *Submitted Jun 5th 2021 at 2:23:17 pm*

Problem 5.1: You are given a hash table with the following backing array, where `?` denotes that a given cell contains an element (regardless of what that element actually is):

0	1	2	3	4	5	6	7	8	9
		?	?		?			?	

What is the probability of having at least 1 collision within the next 3 insertions using linear probing as the collision resolution strategy?

Enter your answer as a decimal rounded to 3 decimal places. For example, if you believe the answer is `12.35%`, enter your answer as `0.124`.

Note: Any filled slots encountered during the probing step of linear probing do not count as collisions: only the initial "hashed to an already-filled slot" event.

0.88

Question 2 *Submitted Jun 5th 2021 at 2:23:19 pm*

Problem 5.2: You are given a hash table with the following backing array, where `?` denotes that a given cell contains an element (regardless of what that element actually is):

0	1	2	3	4	5	6	7	8	9
		?	?		?			?	

What is the probability of having at least 1 collision within the next 3 insertions using separate chaining as the collision resolution strategy?

Enter your answer as a decimal rounded to 3 decimal places. For example, if you believe the answer is `12.35%`, enter your answer as `0.124`.

0.88

Question 3 *Submitted Jun 5th 2021 at 2:23:34 pm*

Problem 5.3: You are given a hash table with the following backing array, where `?` denotes that a given cell contains an element (regardless of what that element actually is):

0	1	2	3	4	5	6	7	8	9
		?	?		?			?	

Using a hash function of $h(x) = 4x+3$, and using linear probing as the collision resolution strategy, if you try to insert the element **5**, which index of the backing array will it insert into?

4

Question 4 *Submitted Jun 5th 2021 at 2:23:56 pm*

Problem 5.4: You are given a hash table with the following backing array, where `?` denotes that a given cell contains an element (regardless of what that element actually is):

0	1	2	3	4	5	6	7	8	9
		?	?		?			?	

Using a hash function of $h(x) = 4x+3$, and using separate chaining as the collision resolution strategy, if you try to insert the element **5**, which index of the backing array will it insert into?

3

Question 5 *Submitted Jun 5th 2021 at 2:24:06 pm*

Problem 5.5: True/False: Given a hash table with a backing array of length 10, if you were to use the hash function $h(x) = 4x$, some of the indices of the backing array will never be filled.

- (●) True
- ( ) False

Question 6 *Submitted Jun 5th 2021 at 2:24:16 pm*

Problem 5.6: Which of the following are valid hash functions? Select all that apply.

Note: Assume `time()` returns the current system time.

Note: Assume `randint()` returns a random positive integer.

- [x] $h(x) = 5$
- [x] $h(x) = x$
- [ ] $h(x) =$ `time()`
- [x] $h(x) = (2x+3) \% 7$
- [ ] $h(x) = x +$ `randint()`
- [x] $h(x) = (5x+7) \% 3x$
- [ ] None of the above

Question 7 *Submitted Jun 5th 2021 at 2:25:43 pm*

Problem 5.7: You are given the following Bloom Filter:

0	1	2	3	4	5	6	7	8	9
T	F	F	T	T	F	T	F	T	T

The Bloom Filter uses the following hash functions:

$$h_1(x) = x + 13$$

$$h_2(x) = 5x$$

$$h_3(x) = 2x + 7$$

Which of the following definitely **are not** in the Bloom Filter? Select all that apply.

- [x] 2
- [x] 3
- [ ] 6
- [x] 10

- [ ] All of the above might appear in the Bloom Filter

Question 8 *Submitted Jun 5th 2021 at 2:25:47 pm*

Problem 5.7: You are given the following Bloom Filter:

0	1	2	3	4	5	6	7	8	9
T	F	F	T	T	F	T	F	T	T

The Bloom Filter uses the following hash functions:

$$h_1(x) = x + 13$$

$$h_2(x) = 5x$$

$$h_3(x) = 2x + 7$$

Which of the following definitely **are** in the Bloom Filter? Select all that apply.

- [ ] 1
- [ ] 5
- [ ] 18
- [ ] 42
- [x] None of the above

Question 9 *Submitted Jun 5th 2021 at 2:25:53 pm*

Problem 5.9: You are given the following Count-Min Sketch:

0	1	2	3	4	5	6	7	8	9	10
0	0	1	1	2	0	0	0	0	0	0
0	1	0	1	1	0	0	0	0	0	1
0	0	0	0	0	1	0	1	1	0	0

The Count-Min Sketch uses the following hash functions:

$$h_1(x) = x + 13$$

$$h_2(x) = 5x$$

$$h_3(x) = 2x + 7$$

What is the maximum possible count of the number **1**?

0

Question 10 *Submitted Jun 5th 2021 at 2:25:56 pm*

Problem 5.10: You are given the following Count-Min Sketch:

0	1	2	3	4	5	6	7	8	9	10
0	0	1	1	2	0	0	0	0	0	0
0	1	0	1	1	0	0	0	0	0	1
0	0	0	0	0	1	0	1	1	0	0

The Count-Min Sketch uses the following hash functions:

$$h_1(x) = x + 13$$

$$h_2(x) = 5x$$

$$h_3(x) = 2x + 7$$

What is the maximum possible count of the number **13**?

0

Week 6: Fast String Searching

This section has 10 questions (below). You may need to scroll down to see them.

Report ambiguous questions: https://forms.gle/ZZ1c2eiYNaMRPepi9

Question 1 *Submitted Jun 5th 2021 at 2:43:18 pm*

Problem 6.1: True/False: Given a string *s* that has length *k*, where *k* does *not* include the null termination symbol (`$`), the Burrows-Wheeler Transform of *s* is guaranteed to be longer than the Suffix Array of *s*.

- (●) True
- ( ) False

Question 2 *Submitted Jun 5th 2021 at 2:43:22 pm*

Problem 6.2: How many failure links are in an Aho-Corasick Automaton with 35 nodes?

35

Question 3 *Submitted Jun 5th 2021 at 2:43:25 pm*

Problem 6.3: You construct an Aho-Corasick Automaton from a dataset containing 5 unique words, each of length 6. How many dictionary links will the resulting automaton have?

Note: Unique words can share letters, but the overall words must be different.

0

Question 4 *Submitted Jun 5th 2021 at 2:43:35 pm*

Problem 6.4: What is the Burrows-Wheeler Transform of the string `FRIDAYFINAL`? Use `$` as the string-terminating symbol (i.e., the string + string-terminating symbol would be `FRIDAYFINAL$`).

LNDIY$RFAIFA

Question 5 *Submitted Jun 5th 2021 at 2:43:43 pm*

Problem 6.5: You are given the following Burrows-Wheeler Transform:

```
nnuvlnrawn$ogdeeteo
```

What is the original string from which this Burrows-Wheeler Transform was constructed?

Note: Include the string-termination symbol (`$`) at the end of your string.

nevergonnaletudown$

Question 6 *Submitted Jun 5th 2021 at 2:44:05 pm*

Problem 6.6: Consider the following string:

```
CLASSICAL
```

Generate the LastToFirst (L2F) array from the BWT of this string.

Note: Order the values of the L2F array such that the top item is the first element of the L2F array and the bottom item is the last element of the L2F array.

Note: `CLASSICAL` is the word, NOT the BWT.

6

3

7

5

0

8

1

4

9

2

Question 7 *Submitted Jun 5th 2021 at 2:44:11 pm*

Problem 6.7: You construct an Aho-Corasick Automaton from the following words:

```
Computer
ComputerScience
Science
Scene
Scent
Data
DataScience
```

How many nodes are in the resulting automaton?

38

Question 8 *Submitted Jun 5th 2021 at 2:44:14 pm*

Problem 6.8: How many failure links are in the automaton you constructed in Problem 6.7?

38

Question 9 *Submitted Jun 5th 2021 at 2:44:17 pm*

Problem 6.9: How many dictionary links are in the automaton you constructed in Problem 6.7?

2

Question 10 *Submitted Jun 5th 2021 at 2:44:54 pm*

Problem 6.10: Construct a Suffix Array from the following string:

```
computation
```

What are the values of the suffix array?

Note: Order the values of the suffix array such that the top item is the first element of the suffix array and the bottom item is the last element of the suffix array.

Week 7: Data Compression

This section has 8 questions (below). You may need to scroll down to see them.

Report ambiguous questions: https://forms.gle/ZZ1c2eiYNaMRPepi9

Question 1 *Submitted Jun 5th 2021 at 4:56:50 pm*

Problem 7.1: How many bits are needed to represent an event if there are 25,000 possible outcomes?

15

Question 2 *Submitted Jun 5th 2021 at 4:57:37 pm*

Problem 7.2: Given an `unsigned char n`, which of the following would set the `i`-th bit from the right to 0 and leave all other bits unchanged? Select all that apply.

Note: `i = 0` corresponds to the least-significant (i.e., right-most) bit.

- [ ] `0 & (n >> i)`
- [ ] `n & (1 << i)`
- [ ] `n | (1 << i)`
- [x] `n & ~(1 << i)`
- [ ] `n | ~(1 << i)`
- [ ] None of the above

Question 3 *Submitted Jun 5th 2021 at 4:57:45 pm*

Problem 7.3: Given an `unsigned char n`, which of the following would return the `i`-th bit from the left? Select all that apply.

Note: `i = 0` corresponds to the most-significant (i.e., left-most) bit.

Note: The command must return 1 for true (rather than any non-zero value)

- [ ] `1 | (n >> i)`
- [ ] `0 | (n >> i)`
- [ ] `1 & (n << i)`
- [ ] `0 & (n << i)`
- [ ] `1 & (n >> i)`
- [x] `1 & (n >> (7 - i))`
- [ ] None of the above

Question 4 *Submitted Jun 5th 2021 at 4:59:09 pm*

Problem 7.4: True/False: If you use some coding scheme (not necessarily Huffman Coding) to encode two strings `x` and `y`, if the encoded version of `x` is longer than the encoded version of `y`, `x` must have more information content (i.e., higher Shannon entropy) than `y`.

- ( ) True
- (•) False

Question 5 *Submitted Jun 5th 2021 at 4:59:14 pm*

Problem 7.5: True/False: A Huffman Tree should always be perfectly balanced to produce the shortest possible encoding of any given message.

- ( ) True
- (•) False

Question 6 *Submitted Jun 5th 2021 at 4:59:24 pm*

Problem 7.6: Build a Huffman Tree from the following symbol frequencies using the Huffman Tree Building algorithm discussed in class:

```
A = 25
D = 4
E = 18
I = 10
M = 5
R = 7
S = 9
U = 2
```

- Make the lower-frequency child the left 0 child, and make the higher-frequency child the right 1 child.
- If two nodes have the same frequency, consider the node with the alphabetically smaller symbol to be the smaller node.
- Label internal nodes with the label of their left 0 child.

How many nodes are in the resulting Huffman Tree?

15

Question 7 *Submitted Jun 5th 2021 at 4:59:39 pm*

Problem 7.7: What is the length (in bits) of the longest symbol encoding in the Huffman Tree you constructed in Problem 7.6?

5

Question 8 *Submitted Jun 7th 2021 at 3:28:36 pm*

Problem 7.8: Encode the string `SUMMER` using the Huffman Tree you constructed in Problem 7.6. What is the resulting binary string?

00110110101010101001000

Week 8: Introduction to Graphs

This section has 10 questions (below). You may need to scroll down to see them.

Report ambiguous questions: https://forms.gle/ZZ1c2eiYNaMRPepi9

Question 1 *Submitted Jun 6th 2021 at 10:52:33 am*

Problem 8.1: You are given the following adjacency list of a directed weighted graph:

```
A: [(C, 4), (F, 3)]
B: [(A, 2), (D, 3), (F, 2)]
C: [(A, 5), (D, 6), (F, 1)]
D: [(B, 6), (E, 4)]
E: []
F: [(E, 3)]
```

If you were to represent this same graph using an adjacency matrix instead, how many cells of the resulting matrix would have a `null` value?

Note: The `null` value represents the lack of an edge.

25

Question 2 *Submitted Jun 6th 2021 at 10:57:14 am*

Problem 8.2: The graph in Problem 8.1 is cyclic.

Note: A graph is "cyclic" if it has at least 1 cycle.

- (●) True
- ( ) False

Question 3 *Submitted Jun 6th 2021 at 10:57:29 am*

Problem 8.3: Which of the following statements about performing Breadth First Search (BFS) and Depth First Search (DFS) on a **Binary Search Tree (BST)** are true? Select all that apply.

- [ ] If we start at the root, the nodes of the BST will be visited in the same order using BFS vs. using DFS

- [x] Both BFS and DFS can be implemented without needing to explicitly keep track of which nodes have already been visited to avoid revisiting the same node multiple times
- [ ] BFS would complete the full tree traversal in fewer steps
- [ ] None of the above

Question 4 *Submitted Jun 6th 2021 at 10:57:46 am*

Problem 8.4: Given an adjacency matrix representing a weighted directed graph and given two nodes *u* and *v*, what is the tightest worst-case Big-O time complexity to find the weight of the edge from *u* to *v* (or `null` if there does not exist an edge from *u* to *v*)?

- (•) O(1)
- ( ) O(log $|V|$)
- ( ) O($|V|$)
- ( ) O($|V|$ log $|V|$)
- ( ) O($|V|^2$)
- ( ) None of the above

Question 5 *Submitted Jun 6th 2021 at 10:57:51 am*

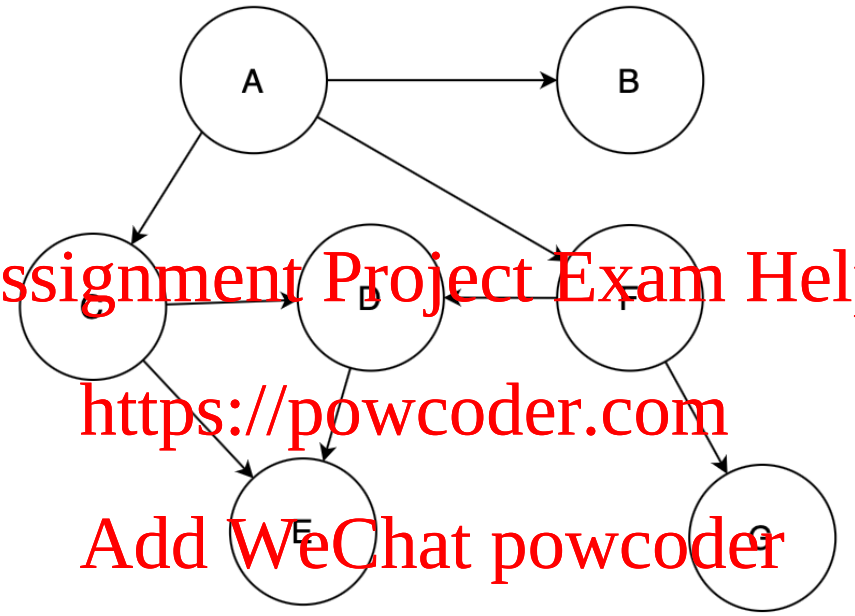
Problem 8.5: Given an adjacency matrix representing a weighted directed graph, what is the tightest worst-case Big-O time complexity to find the total number of edges in the graph?

- ( ) O(1)
- ( ) O(log $|V|$)
- ( ) O($|V|$)

- O(|V| log |V|)
- **O(|V|^2)**
- None of the above

Question 6 *Submitted Jun 6th 2021 at 10:58:19 am*

Problem 8.6: You are given the following directed unweighted graph:



Starting at node A, in which order will the nodes in the graph be visited by Breadth First Search (BFS)?

Note: If you encounter a node with multiple neighbors, you should visit them in alphabetical order.

Note: Order the items such that the first node to be visited is on top, and the last node to be visited is on the bottom.

A

B

C

F

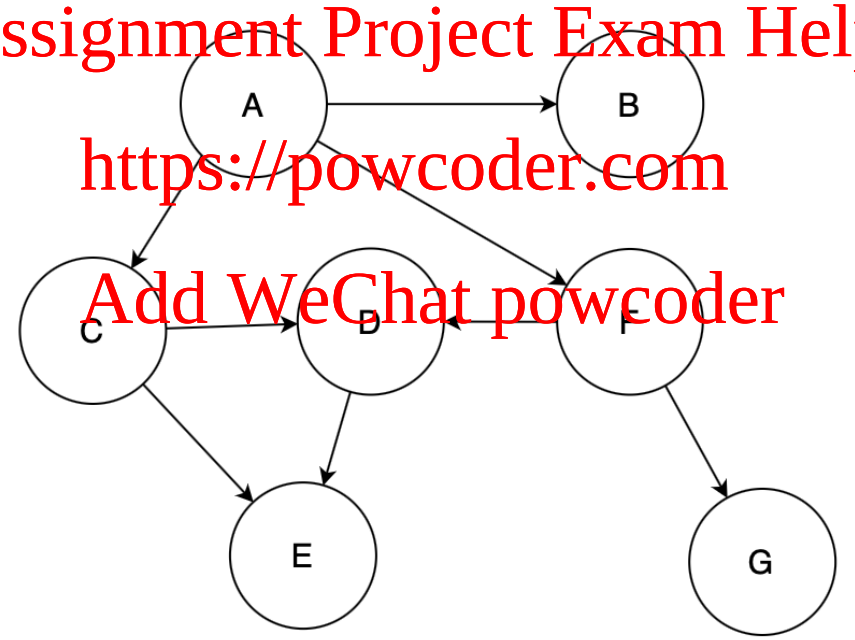
D

E

G

Question 7 *Submitted Jun 6th 2021 at 10:58:37 am*

Problem 8.7: You are given the following directed unweighted graph:



Starting at node A, in which order will the nodes in the graph be visited by Depth First Search (DFS)?

Note: If you encounter a node with multiple neighbors, you should visit them in alphabetical order.

Note: Order the items such that the first node to be visited is on top, and the last node to be visited is on the bottom.

A

B

C

D

E

F

G

Question 8 *Submitted Jun 7th 2021 at 2:09:05 pm*

Problem 8.8: Which of the following statements about Dijkstra's algorithm are true? Select all that apply.

- [x] Dijkstra's algorithm does not support negative edge weights.
- [ ] Dijkstra's algorithm requires the graph to be acyclic
- [x] Dijkstra's algorithm can be used on an unweighted graph
- [x] Dijkstra's algorithm can be used on a tree
- [ ] None of the above

Question 9 *Submitted Jun 6th 2021 at 10:58:53 am*

Problem 8.9: Which of the following statements about Breadth First Search (BFS) and Depth First Search (DFS) are true? Select all that apply.

- [ ] DFS can be used to find a shortest unweighted path
- [x] BFS can be used to find a shortest unweighted path

- [ ] BFS can be used to find a shortest weighted path
- [x] BFS and DFS have the same tightest worst-case Big-O time complexity
- [ ] None of the above

Question 10 *Submitted Jun 6th 2021 at 10:59:01 am*

Problem 8.10: When performing a graph traversal algorithm like Breadth First Search (BFS), Depth First Search (DFS), or Dijkstra's algorithm, the tightest worst-case Big-O time complexity of the algorithm when representing the graph using an adjacency list will be faster than the tightest worst-case Big-O time complexity of the algorithm when representing the graph using an adjacency matrix.

- (•) True
- ( ) False

Week 9: Advanced Graph Algorithms

This section has 8 questions (below). You may need to scroll down to see them.

Report ambiguous questions: https://forms.gle/ZZ1c2eiYNaMRPepi9

Question 1 *Submitted Jun 6th 2021 at 11:54:12 am*

Problem 9.1: True/False: Starting from the same node, Prim's algorithm and Dijkstra's algorithm are guaranteed to visit the nodes of a graph in the exact same order.

- ○ True
- ◉ False

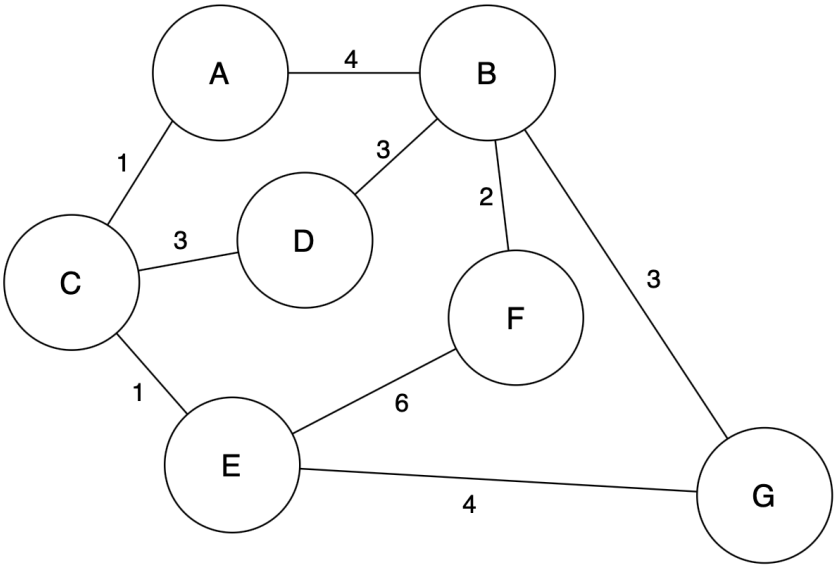
Question 2 *Submitted Jun 6th 2021 at 11:54:15 am*

Problem 9.2: True/False: In a graph in which every edge weight is unique (i.e., no two edges have the same weight), Prim's algorithm and Kruskal's algorithm are guaranteed to yield the exact same Minimum Spanning Tree.

- ◉ True
- ○ False

Question 3 *Submitted Jun 6th 2021 at 11:58:23 am*

Problem 9.3: You are given the following undirected weighted graph:



Starting at node C, in which order will the nodes in the graph be visited by Prim's algorithm?

Note: Break ties by visiting nodes in alphabetical order.

Note: Order the items such that the first node to be visited is on top, and the last node to be visited is on the bottom.

C

A

E

D

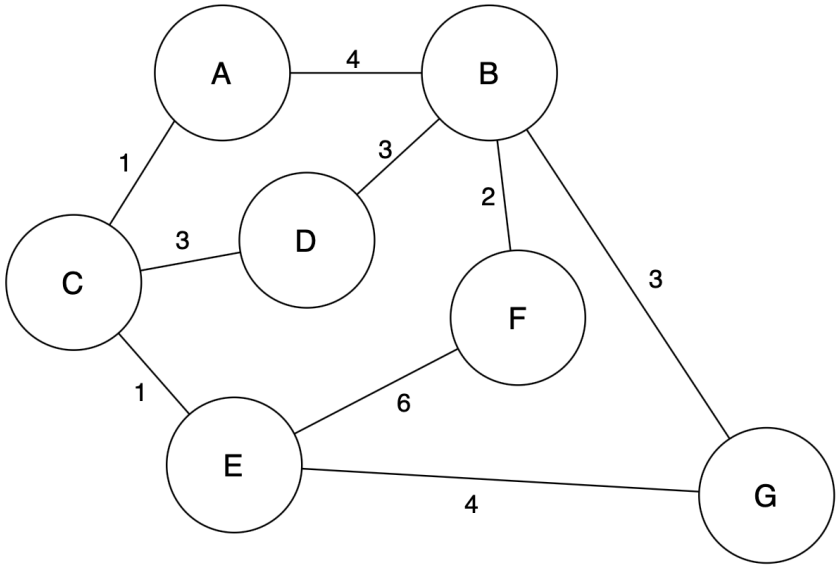
B

F

G

Question 4 *Submitted Jun 6th 2021 at 11:59:44 am*

Problem 9.4: You are given the following undirected weighted graph:



In which order will edges be added to the output Minimum Spanning Tree by Kruskal's algorithm?

Note: All edges are undirected, so we are listing them with the alphabetically-smaller node first. For example, we would write (X,Y,4) instead of (Y,X,4).

*Note: If you have a tie between edges (u,v,weight) and (x,y,weight), break ties alphabetically by the first node, and then by the second node. For example, (**X**,Y,4) would go before (**Y**,Z,4) because X < Y, and (X,**Y**,4) would go before (X,**Z**,4) because Y < Z.*

Note: Order the items such that the first edge to be added is on top, and the last edge to be added is on the bottom.

Note: You will not necessarily include all edges in your answer.

Drag from here

(E,G,4)

(E,F,6)

(A,B,4)

Drop blocks here

(A,C,1)

(C,E,1)

(B,F,2)

(B,D,3)

(B,G,3)

(C,D,3)

Question 5 *Submitted Jun 6th 2021 at 12:00:04 pm*

Problem 9.5: In Kruskal's algorithm, when we're considering the next smallest edge, we want to be able to quickly check that adding this edge into our growing Minimum Spanning Tree will not cause a

cycle. Which data structure could we use to quickly perform a "Will this edge cause a cycle?" check?

- ○ Heap
- ◉ Up-Tree
- ○ Multiway Trie
- ○ Array List
- ○ None of the above

Question 6 *Submitted Jun 6th 2021 at 12:00:15 pm*

Problem 9.6: Which of the following definitions of an Up-Tree's `union(x,y)` operation are guaranteed to result in a valid Up-Tree? Select all that apply.

- ☐ Make `x` point to the sentinel of `y`
- ☑ Make the sentinel of `x` point to `y`
- ☑ Make the sentinel of the smaller set point to the sentinel of the larger set
- ☑ Make the sentinel of `y` point to the sentinel of `x`
- ☐ None of the above

Question 7 *Submitted Jun 6th 2021 at 12:05:02 pm*

Problem 9.7: Imagine we start with an unconnected forest of nodes labeled 0 through 7. Then, we perform the following operations:

```
union(1,4)
union(2,3)
union(5,1)
union(2,5)
union(0,7)
union(0,6)
union(2,6)
```

Assuming we are using Union-by-Size and assuming `find` performs path compression, what is the array representation of our resulting disjoint set?

Provide the array representation as space-separated numbers. For example, if you wanted to represent the initial forest of nodes, your answer would look like the following:

```
-1 -1 -1 -1 -1 -1 -1 -1
```

Note: If you perform `union(x,y)` and the two sets have the exact same size, make the sentinel of `x` point to the sentinel of `y`.

7 4 4 4 -1 4 7 4

Question 8 *Submitted Jun 6th 2021 at 12:05:06 pm*

Problem 9.8: You are given a Disjoint Set that is implemented using an Up-Tree that uses Union-by-Height but does not use Path Compression. You know that the Disjoint Set contains *n* elements in a single set (i.e., there are *n* total nodes and 1 sentinel node), but you do not know the order of operations that produced this Disjoint Set.

What is the tightest worst-case Big-O time complexity to perform a "find" operation on this Disjoint Set?

- O(1)
- **O(log *n*)**
- O(*n*)
- O(*n* log *n*)
- O(n^2)
- None of the above

Week 10: Finishing Up

This section has 2 questions (below). You may need to scroll down to see them.

Report ambiguous questions: https://forms.gle/ZZ1c2eiYNaMRPepi9

Question 1 *Submitted Jun 6th 2021 at 12:09:31 pm*

Problem 10.1: True/False: To prove that a problem is NP-Hard, you must prove that it cannot be verified in polynomial time.

- ○ True
- ◉ False

Question 2 *Submitted Jun 6th 2021 at 12:09:38 pm*

Problem 10.2: What is Niema's favorite video game?

- ◉ Final Fantasy VII
- ○ Final Fantasy VIII
- ○ Final Fantasy IX
- ○ Final Fantasy X

Submission Info

For each of the exam problems, you have to click the "Submit" button to submit your response, and you have **unlimited attempts** without penalty.

However, **you will not know if your answer is correct!** Ed will tell you whether or not you have submitted, but we will grade the correctness of your responses after the exam.

There is no global "Submit" button: as long as you've clicked the "Submit" button for each individual question, your answers are submitted.

Do not forget to click the "Submit" button on every single question!!!