Midterm 1

● Graded

Student

AKSHAJ KAMMARI

Total Points

55 / 150 pts

Question 1

Arrays **11** / 30 pts

1.1 **(a)** Resolved **3** / 10 pts

+ 5 pts (n-1) * n/2

+ 5 pts n^2 – n / 2

+ 5 pts (n -1) * (n-1+1)/2

+ 5 pts (n-1) + (n-2) + ... + 1

+ 5 pts Correct explanation

✓ **+ 0 pts** Incorrect

+ 3 pts Point adjustment

Regrade Request **Submitted on: Mar 14**

Even though my initial answer itself may be incorrect, I think that my explanation still matches up pretty well with the correct one so I think it's fair for me to get partial credit for the explanation

Partial credit for partially correct explanation

Reviewed on: Mar 24

1.2 **(b)** **0** / 10 pts

+ 5 pts (2 * (n-1) * n/2)

if statement in the inner for loop has 2 array accesses

+ 5 pts 4 * (n-1)

4 array accesses in the outer for loop, it executes n-1 times

✓ **+ 0 pts** Incorrect

1.3 **(c)** **Resolved** 3 / 5 pts

+ 5 pts ~n^2

✓ **+ 0 pts** Incorrect

+ 3 pts Point adjustment

Regrade Request **Submitted on: Mar 14**

I'm hoping I'm able to get credit back for this problem, or at least partial credit, because I got the content of what the bulk of the answer was and I believe that's more important than what I missed.

~ missing. Partial credit given.

Reviewed on: Mar 19

1.4 **(d)** **5** / 5 pts

✓ **+ 5 pts** $O(n^2)$

+ 0 pts Incorrect

Question 2

Union-Find **11** / 30 pts

2.1 (a) **4** / 10 pts

+ 3 pts

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

Indices 0, 1, 5, and 7 have the SAME value (either 0, 1, 5, or 7)

✓ **+ 4 pts**

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

Indices 4, 6, and 9 have the value 4, 6, 9 respectfully

+ 3 pts

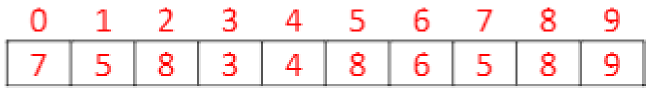
0	1	2	3	4	5	6	7	8	9
5	5	3	3	4	5	6	5	3	9

Indices 2, 3, and 8 have the SAME value (either 2, 3, or 8)

+ 0 pts Incorrect

2.2 **(b)** **6** / 10 pts

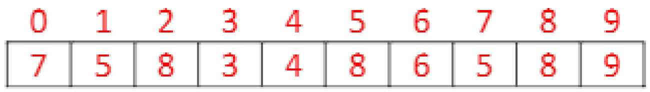
+ 2 pts



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

Indices 1 and 7 have the same value 5

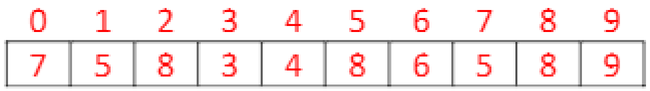
+ 2 pts



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

Index 0 has the value 7

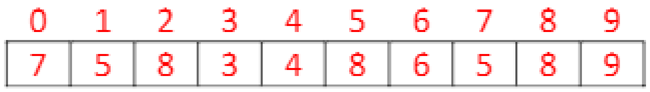
✓ **+ 3 pts**



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

Indices 2, 5, and 8 have the value 8

✓ **+ 3 pts**



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

Indices 3, 4, 6, and 9 separately have the value 3, 4, 6, 9 respectfully

+ 0 pts Incorrect

2.3 **(c)** 1 / 10 pts

+ 3 pts parent[]

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

Indices 0, 1, 2, and 7 have the value 0

+ 1 pt parent[]

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

Index 5 has the value 1, index 8 has the value 2

✓ **+ 1 pt** parent[]

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

Indices 3, 4, 6, and 9 separately have the value 3, 4, 6, and 9

+ 3 pts size[]

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

Index 0 has 6

+ 1 pt size[]

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

Indices 1 and 2 have 2

+ 1 pt size[]

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

All others have 1

+ 0 pts Incorrect

Question 3

Stacks and Queues **22** / 30 pts

3.1 (a) **0** / 8 pts

+ 8 pts 3 5 4
16
0

✓ **+ 0 pts** Incorrect

3.2 (b) **8** / 8 pts

✓ **+ 8 pts** eueuq dna kcats

+ 0 pts Incorrect

3.3 (c.1) **7** / 7 pts

✓ **+ 2 pts** No

✓ **+ 5 pts** The for-loop push (add) the count values in ascending order, and the while loop pop (remove) the values in reverse order; the output 135 is possible, however, the output 24 is not possible.

+ 0 pts Incorrect

3.4 (c.2) **7** / 7 pts

✓ **+ 2 pts** Yes

✓ **+ 5 pts** The output is generated when the odd values are printed directly and the even values are pushed to the stack. The even values will be displayed in a reverse order.

+ 0 pts Incorrect

Question 4

Linked Lists and Arrays **4** / 30 pts

4.1 (a) Resolved 2 / 15 pts

+ 2 pts
```
private boolean isPresent (Color color) {

    Node ptr = uniqueColorList;           // starts at front [2 points]
    while ( ptr != null ) {               // correct loop cond [4 points]
        if ( ptr.pixel.equals(color) ) { // checks equality [5 points]
            return true;                  // returns true on equal [2 points]
        }
    }
    return false;                         // returns false on !equal [2 points]
}
```

Starts at front
Node ptr = uniqueColorList;

+ 4 pts
```
private boolean isPresent (Color color) {

    Node ptr = uniqueColorList;           // starts at front [2 points]
    while ( ptr != null ) {               // correct loop cond [4 points]
        if ( ptr.pixel.equals(color) ) { // checks equality [5 points]
            return true;                  // returns true on equal [2 points]
        }
    }
    return false;                         // returns false on !equal [2 points]
}
```

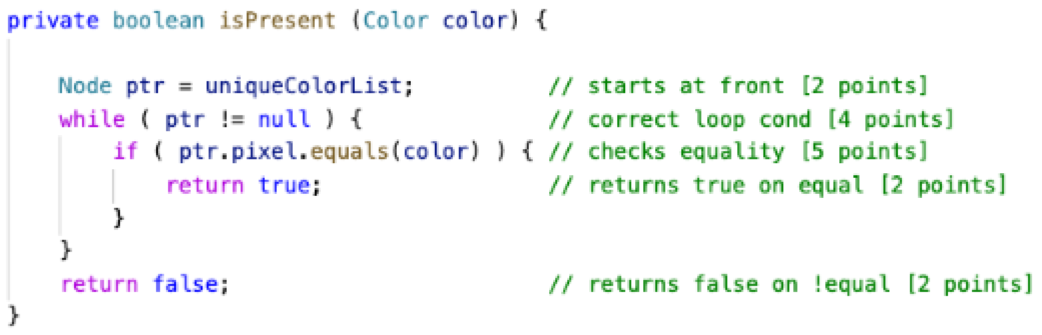
Correct loop condition
while(ptr != null)

+ 5 pts
```
private boolean isPresent (Color color) {

    Node ptr = uniqueColorList;           // starts at front [2 points]
    while ( ptr != null ) {               // correct loop cond [4 points]
        if ( ptr.pixel.equals(color) ) { // checks equality [5 points]
            return true;                  // returns true on equal [2 points]
        }
    }
    return false;                         // returns false on !equal [2 points]
}
```

Checks equality
if(ptr.pixel.equals(color))

✓ + 2 pts
```
private boolean isPresent (Color color) {

    Node ptr = uniqueColorList;           // starts at front [2 points]
    while ( ptr != null ) {               // correct loop cond [4 points]
        if ( ptr.pixel.equals(color) ) { // checks equality [5 points]
            return true;                  // returns true on equal [2 points]
        }
    }
    return false;                         // returns false on !equal [2 points]
}
```

Return true on equal
return true;

+ 2 pts 

```
private boolean isPresent (Color color) {

    Node ptr = uniqueColorList;          // starts at front [2 points]
    while ( ptr != null ) {              // correct loop cond [4 points]
        if ( ptr.pixel.equals(color) ) { // checks equality [5 points]
            return true;                 // returns true on equal [2 points]
        }
    }
    return false;                        // returns false on !equal [2 points]
}
```

Returns false on not equal

return false

+ 0 pts Incorrect

Regrade Request **Submitted on: Mar 14**

Even though I didn't get the code right, I think I technically did have a return true on an equals and i was told that we would not lose points for syntax mistakes?

I will consider this, but uniqueColorList is a linked list, not an array

Reviewed on: Mar 19

4.2 (b) Resolved 2 / 15 pts

+ 2 pts

```
private void insertFront (Color color) {
    // create new node with pixelColor
    Node newNode = new Node();              // creates new node [5 points]
    newNode.pixel = color;                  // assigns color to pixel [5 points]

    // insert at the front of the list
    newNode.next = uniqueColorList;       // insert new node at front [5 points]
    uniqueColorList = newNode;            // update front [5 points]
}
```

Creates new node

Node newNode = new Node();

+ 3 pts

```
private void insertFront (Color color) {
    // create new node with pixelColor
    Node newNode = new Node();              // creates new node [5 points]
    newNode.pixel = color;                  // assigns color to pixel [5 points]

    // insert at the front of the list
    newNode.next = uniqueColorList;       // insert new node at front [5 points]
    uniqueColorList = newNode;            // update front [5 points]
}
```

Assigns color to pixel

newNode.pixel = color;

+ 5 pts

```
private void insertFront (Color color) {
    // create new node with pixelColor
    Node newNode = new Node();              // creates new node [5 points]
    newNode.pixel = color;                  // assigns color to pixel [5 points]

    // insert at the front of the list
    newNode.next = uniqueColorList;       // insert new node at front [5 points]
    uniqueColorList = newNode;            // update front [5 points]
}
```

Insert new node at front

newNode.next = uniqueColorList

+ 5 pts

```
private void insertFront (Color color) {
    // create new node with pixelColor
    Node newNode = new Node();              // creates new node [5 points]
    newNode.pixel = color;                  // assigns color to pixel [5 points]

    // insert at the front of the list
    newNode.next = uniqueColorList;       // insert new node at front [5 points]
    uniqueColorList = newNode;            // update front [5 points]
}
```

Update front

uniqueColorList = newNode;

✓ + 0 pts Incorrect

+ 2 pts Point adjustment

Regrade Request — Submitted on: Mar 14

I did most of this problem wrong but I did assign it at the front so I was wondering if maybe I could get the partial credit for that?

I will give credit for an attempt, but that is all I can do.

Reviewed on: Mar 19

Question 5

Binary Search Tree (BST) — **7** / 30 pts

5.1 (a) — **0** / 7 pts

+ 7 pts 5 comparisons

✓ + 0 pts Incorrect

5.2 (b) — **0** / 8 pts

+ 3 pts 2 comparisons

+ 5 pts Average number of comparisons = (2+2+3+1+2+2+2)/7 = 2

Search fails for 1, 2, 11, 20: 2 comparisons each
Search succeeds for 6, 7, 9: 3, 1, 2 comparisons respectively

✓ + 0 pts Incorrect

5.3 (c) — Resolved — **0** / 8 pts

+ 3 pts O(n)

+ 5 pts In the worst case, the tree would be skewed to the right with the largest element being the only leaf node on the right side. Therefore, you would have to iterate through all the elements to get to the largest element.

✓ + 0 pts Incorrect

Regrade Request — Submitted on: Mar 14

I understand that my answer is incorrect and why I lost the 3 points, but I think that my explanation would still go hand in hand with the correct answer and that my logic behind it is still valid.

Sorry, cannot give credit if part a is not right.

Reviewed on: Mar 15

5.4 (d) — **7** / 7 pts

✓ + 7 pts D. 10, 7, 15, 20, 2, 8

+ 0 pts Incorrect

CS112 Data Structures - Midterm 1 - Spring 2022

Name: Akshaj Kammari NetID: AK1990

- **WRITE your name and NetID on EVERY page.**
- **DO NOT REMOVE** THE STAPLE IN YOUR EXAM.
- **DO NOT BEGIN** UNTIL INSTRUCTED TO DO SO.
- WRITE NEATLY AND CLEARLY. If we cannot read your handwriting, you will not receive credit. Please plan your space usage. No additional paper will be given.
- This exam is worth 150 points.

Problem 1 – Arrays (30 points)

Given the code segment below.

```
for ( int i = 0; i < n - 1; i++ ) {

    int min = i;

    for ( int j = i + 1; j < n; j++ ) {

        if ( a[j] < a[min] ) {
            min = j;
         }
    }

    int temp = a[i];
    a[i] = a[min];
    a[min] = temp;
}
```

(a) **(10 points)** How many times is the `if` statement in the inner `for` loop executed? Give your answer as a function of n with a succinct explanation.

$$\frac{(n-1)^2}{2}$$

The outer for loop will only run n-1 times because of the statement "$i < n-1$" which is multiplied by n-1 because it will run the same number of times and is nested. It divided by 2 because of the number of times it is iterated

Name: Akshaj Kammari NetID: AK1990

(b) **(10 points)** What is the maximum number of array accesses (reads and writes) for the entire code segment as a function of n? Justify your answer.

$2(n-1)^2+4$

There are 2 accesses inside of the 'if' statement which is run $(n-1)^2$ times, thus $2(n-1)^2$, and there are 4 accesses outisíde of the loops which justify the +4.

(c) **(5 points)** Write the tilde notation for the function in (b).

n^2

(d) **(5 points)** Write the Big-O notation for the function in (b).

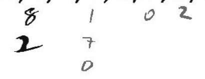
$O(n^2)$

CS112 Data Structures - Midterm 1 - Spring 2022

Name: Akshaj Kammari NetID: AK1990

Problem 2 – Union-Find (30 points)

A client program is using the union-find API to solve a dynamic connectivity problem in networking. Assume that there are 10 sites identified as: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 in the network. Answer the following questions.

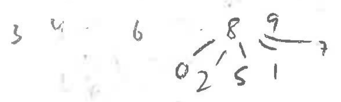


(a) **(10 points)** If the union-find API is implemented with the quick-find discussed in class, show the content of the `id[]` array after adding the following edges in sequence: 0-7, 1-5, 2-8, 7-1, 8-3

	0	1	2	3	4	5	6	7	8	9
id []	7	5	8	3	4	5	6	1	3	9

(b) **(10 points)** If the API is implemented with quick-union discussed in class, show the content of the `parent[]` array after adding the following edges in sequence: 0-7, 1-5, 2-8, 7-1, 5-2

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



Name: Akshaj kammari NetID: AK1990

(c) **(10 points)** If the API is implemented with weighted-quick-union discussed (union by size) in class, show the content of the parent[] and size[] array after adding the same edges from part (b).

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

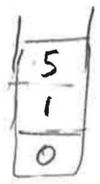
0	1	2	3	4	5	6	7	8	9
7	5	8	3	4	5	6	1	3	9

Name: Akshaj Kammari NetID: AK 1990

Problem 3 – Stacks and Queues (30 points)

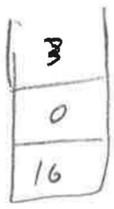
stack - filo

(a) **(8 points)** What is the output of the following code segment?



```
Stack<Integer> stack = new Stack<Integer>();
int item1 = 1;
int item2 = 0;
int item3 = 4;

stack.push( item2 );
stack.push( item1 );
stack.push( item1 + item3 );

item2 = stack.pop();

stack.pop();
stack.push( item3 * item3 );
stack.push( item2 );
stack.push( 3 );

item1 = stack.pop();

stack.pop();
System.out.println( item1 + " " + item2 + " " + item3 );

while ( !stack.empty() ) {
    System.out.println( stack.pop() );
}
```

5 10 30 16



Answer:

5 1 0 3 0 16

Name: Akshaj Kammari NetID: AK1990

(b) **(8 points)** What is the output of the following code segment?

fifo

```
 Queue<Character> queue = new Queue<Character>();
Stack<Character> stack = new Stack<Character>();

String s = "stack and queue";

char [] charArray = s.toCharArray();

for ( int i = 0; i < charArray.length; i++ ) {
    stack.push( charArray[i] );
}

while ( !stack.isEmpty() ) {
    queue.enqueue(stack.pop());
}

while ( !queue.isEmpty() ) {
    System.out.print( queue.dequeue() );
}
```

Answer:

e u e u q   d n a   k c a t s

Name: Akshaj Kammari NetID: AK1990

(c) Answer questions based in the code segment below. Note that `randomBoolean()` randomly returns `true` or `false`.

```
Stack<Integer> stack = new Stack<>();

for ( int count = 1; count <= 5; count++ ) {

    if ( randomBoolean() ) {
        System.out.print( count );
    } else {
        stack.push( count );
    }
}

while ( !stack.isEmpty() ) {
    System.out.print( stack.pop() );
}
```

1. **(6 points)** Is the output 13524 possible? Explain.

No, because of FILO 4 must come before 2 while the stack is popped.

2. **(6 points)** Is the output 13542 possible? Explain.

Yes, because the boolean can be true when count is 1, 3, and 5 and 2 and 4 can be sent to stack and which will be printed out as 42 after 135 due to FILO.

CS112 Data Structures - Midterm 1 - Spring 2022

Name: Akshaj Kammari NetID: Ak1990

Problem 4 – Linked Lists and Arrays (30 points)

On ArtCollage you worked with the Picture class which follow the digital image abstraction. The Picture is a 2D array of Color values, see operations below.

```
public class Picture

              Picture(String filename)              create a picture from a file
              Picture(int w, int h)                 create a blank w-by-h picture
         int  width()                               return the width of the picture
         int  height()                              return the height of the picture
       Color  get(int col, int row)                 return the color of pixel (col, row)
        void  set(int col, int row, Color color)    set the color of pixel (col, row) to color
        void  show()                                display the picture in a window
        void  save(String filename)                 save the picture to a file
```

The ImageProcessing class below also uses the Picture class, it contains two instance variables:

- ***image***: a reference to a Picture.
- ***uniqueColorList***: a reference to the front node of a linked list that stores all the unique colors from the image.

The class also contains the methods:

- ***populateUniqueColorList***: that inserts in the *uniqueColorList* all pixel colors that are unique. The list will NOT contain duplicate colors.
- ***isPresent(Color color)***: returns true if the parameter *color* is present in the *uniqueColorList*.
- ***insertFront(Color color)***: creates a new linked list node where *pixel* refers to the color of the pixel, and inserts the newly created node at the front of the *uniqueColorList*.

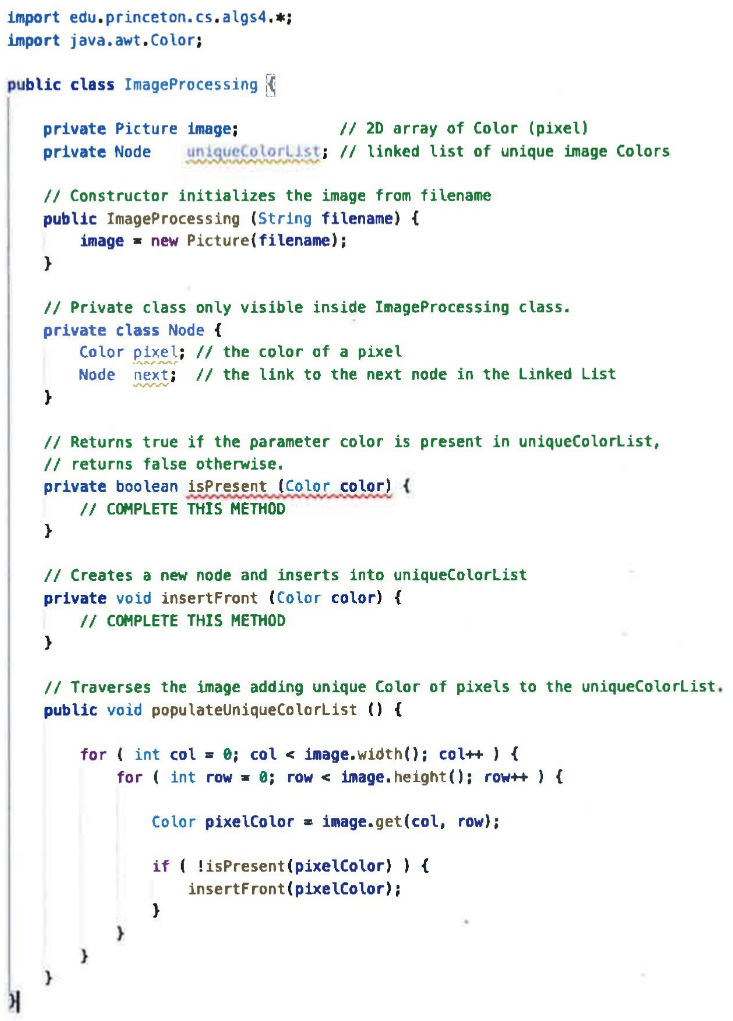
a) **(15 points)** Implement the ***isPresent*** method.

```
private boolean isPresent (Color color) {
for (int i = 0 ; i<uniqueColorList.length; uniqueColorList.next) {
if (color == uniqueColorList[i]) {
  return true;
  }
```

Name: Akshaj Kammari NetID: AK 1990

b) **(15 points)** Implement the ***insertFront*** method.

```
private void insertFront (Color color) {
Node pixel;
temp = uniqueColorList.front;
pixel = uniqueColorList.front;
temp = pixel.next;
```



```
import edu.princeton.cs.algs4.*;
import java.awt.Color;

public class ImageProcessing {

    private Picture image;            // 2D array of Color (pixel)
    private Node    uniqueColorList; // linked list of unique image Colors

    // Constructor initializes the image from filename
    public ImageProcessing (String filename) {
        image = new Picture(filename);
    }

    // Private class only visible inside ImageProcessing class.
    private class Node {
        Color pixel; // the color of a pixel
        Node  next;  // the link to the next node in the Linked List
    }

    // Returns true if the parameter color is present in uniqueColorList,
    // returns false otherwise.
    private boolean isPresent (Color color) {
        // COMPLETE THIS METHOD
    }

    // Creates a new node and inserts into uniqueColorList
    private void insertFront (Color color) {
        // COMPLETE THIS METHOD
    }

    // Traverses the image adding unique Color of pixels to the uniqueColorList.
    public void populateUniqueColorList () {

        for ( int col = 0; col < image.width(); col++ ) {
            for ( int row = 0; row < image.height(); row++ ) {

                Color pixelColor = image.get(col, row);

                if ( !isPresent(pixelColor) ) {
                    insertFront(pixelColor);
                }
            }
        }
    }
}
```

CS112 Data Structures - Midterm 1 - Spring 2022

Name: Akshaj kammari NetID: AK1990

Problem 5 – Binary Search Tree (BST) (30 points)

(a) **(7 points)** After inserting 18, 51, 37, 11, 46, 25, and 20 into an empty BST in that order, what would be the worst case number of comparisons (compareTo calls) for a successful search?

O log n

(b) **(8 points)** If we perform searches for 1, 2, 6, 7, 9, 11, 20 in the following BST, what would be the average number of comparisons (compareTo calls), regardless of whether the search ends in success or failure? Give the reasoning.

```
  7
 / \
4   9
 \
  6
```

3 to 4

(c) **(8 points)** In the worst case scenario, what would be the time complexity to delete the node containing the largest value in a BST? Give the reasoning.

O log n , because it would have to traverse through the whole tree because the largest value would be all the way to the right.

(d) **(7 points)** Given the following numbers to insert into an empty BST: 2, 7, 8, 10, 15, 20. What insertion order would yield the tree with the least height?

~~A~~. 15, 2, 20, 8, 7, 10

~~B~~. 8, 20, 7, 2, 15, 10

~~C~~. 7, 2, 10, 8, 15, 20

[✓ D]. 10, 7, 15, 20, 2, 8