Prince Sultan University
College of Computer and Information Sciences
Department of Computer Science

Midterm Exam
Term 211

Course title: Data Structures **Course Code**: **CS 210**

Exam date: 10 November 2021 **Exam Time: 50** minutes

Student Name:

Student ID:

Circle your instructors name

Dr. Basit Qureshi Dr. Mohamed Akour Dr. Syed Umer Amin Dr. Sakhar Alkhereyf

Dr. Sawsan Alhalawani Dr. Siwar Rekik Dr. Najla Althuniyan

Question Number	CLO	Question points	Score
Question 1	**CLO 1**	10	
Question 2	**CLO 1**	5	
Question 3	**CLO 3**	5	
Total out of		**20**	

Instructions:

- This exam contains three questions with multiple parts.
- Time allowed: 50 minutes
- Closed Book, Closed Notes.
- Use of Calculators and / or computing devices / smartphones etc is strictly prohibited.
- Answer the problems on the exam sheets only. No additional attachments would be accepted.
- If you need extra space use the back of a page.
- When the “time is over” is called, it is students’ responsibility to submit his exam to the invigilator. Submitting completed exam 3 minutes after the “time is over” will incur a penalty of **5 points**.

Few gentle reminders:

- If you get stuck on some problem for a long time, move on to the next one.
- The ordering of the problems is somewhat related to their relative difficulty. However, the order might be different for you!
- You should be better off by first reading all questions and answering them in the order of what you think is the easiest to the hardest problem.
- Keep the points distribution in mind when deciding how much time to spend on each problem.

Question 1 [/10 marks]

Part a: [4 points] Estimate the run time as T(n) and then state the runtime requirements in big-O for each of the following code fragments.

	Code Fragment	T(n)	Running Time in big-O

a	int a = 0;
	for (i = 0; i < N; i++) {
	    for (j = N; j > i; j--) {
	        a = a + i + j;
	    }
	}		

b	int i, j, k = 0, m=0;
	
	for(i=0; i<=n; i++)
	    m = m + m * i;
	
	for (i = n / 2; i <= n; i++) {
	    for (j = 2; j <= n; j = j * 2) {
	        k = k + n / 2;
	    }
	}		

c	public void print(int n){
	  for (int i = 1; i <= n; i++) n
	      System.out.println(i); 1
	}
	
	public void main(){
	  Scanner Key= new Scanner(System.in); 3
	  int n=Key.nextInt(); 3
	
	    for (i = 1; i <= n; i++) n
	         for (int j = 1; j <= n; j++) n
	             print(i+j); 1 + n
	}		

d	public void square(int n){
	     return n*n; 2
	
	}
	
	public void main(){
	  Scanner Key= new Scanner(System.in); 3
	  int n=Key.nextInt(); 3
	
	  int s = square(n); 3
	}		

Part b: [2 points] For what values of n_o and constant c, the function
f(n) **=** $\mathbf{n^4 + 36n^2}$ **is** $\mathbf{O(n^4)}$

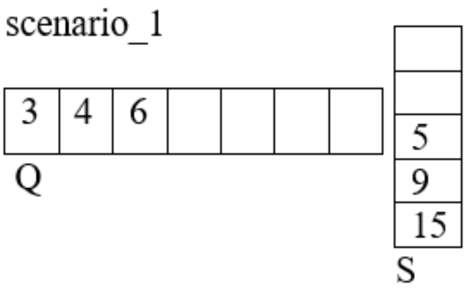
from the definition of big-O
n^4+36n^2 is $O(n^4)$ if $n^4+36n^2 <= c. n^4$ for all n>n0
assume c = 2,
$n^4+36n^2 <= 2. n^4$
$36n^2<= n^4$
divide over n^2
$36<=n^2$ true for n>=6
So, n^4+36n^2 is $O(n^4)$ for c=2 and n>=6

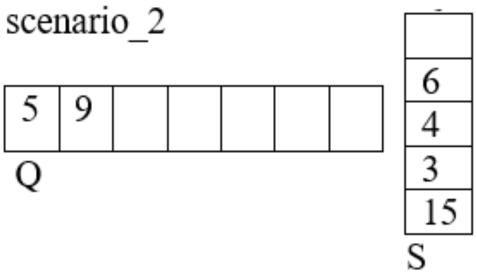
Part c [4 points]: For each of the following scenarios determine the faster run time (less time)

Who is telling the truth?	Ahmed or Saleh?	Briefly explain why?
Ahmed says: Inserting an element at the beginning of a circular linked list (vs) Saleh says: Inserting an element in an AVL tree	Ahmed	Insertion in a circular linked list takes O(1) while in an AVL tree takes O(logn)
Ahmed says: Searching a value in an array using binary search (items are sorted) (vs) Saleh says: Searching a value in a BST	Ahmed	Searching in an array using binary search takes O(logn) while in a BST takes O(n) in the worst case if the values are sorted when they were inserted in the BST
Ahmed says: Searching and removing a value from a Queue (vs) Saleh says: Searching and removing a value from an AVL Tree	Saleh	In AVL: it takes O(logn) while in a queue is O(n)
Ahmed says: Post-Order Traversal is faster on a BST. (vs) Saleh says: Post-Order Traversal is faster on a AVL Tree.	If a student wrote AVL (Saleh gives the right answer) as the obvious answer, we can give him/her full mark for this. .	If the student suggested that BOTH gave the correct answer (or both are wrong); we can see the explanation provided. If he/she states that the Post-Order traversal would yield O(n) time regardless of the BST/AVL tree, he/she can be given +1 bonus

Question 2 [/5 marks]

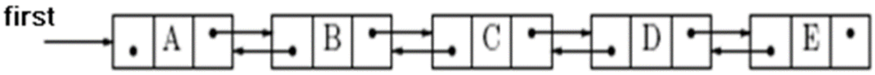
Part a [3 points]: Assume you have a non-empty Stack S and a Queue Q of type integers. Provide a sequence of calls to modify the contents of S and Q from scenario_1 to scenario_2.





```
Q.enqueue( S.pop() ) ;
Q.enqueue( S.pop() ) ;
S.push( Q.dequeue() );
S.push( Q.dequeue() );
S.push( Q.dequeue() );
```

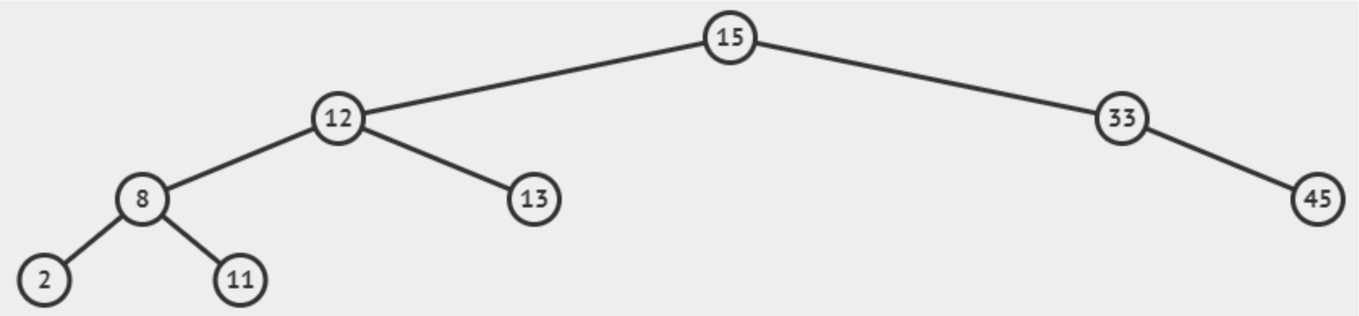
Part b [2 points]: Consider the following double linked list, where the next pointer points to the next node and the back pointer points to the previous node. Write a Java code to delete the third node **(with value C)** in the list using the first pointer. Just write the relevant lines of code. No need to write the entire method.



```
Node<E> first = header.getNext();
Node<E> node = first.getNext().getNext();

Node<E> predecessor = node.getPrev();
Node<E> successor.getNext();

predecessor.setNext(successor);
successor.setPrev(predecessor);

size-- ;
```

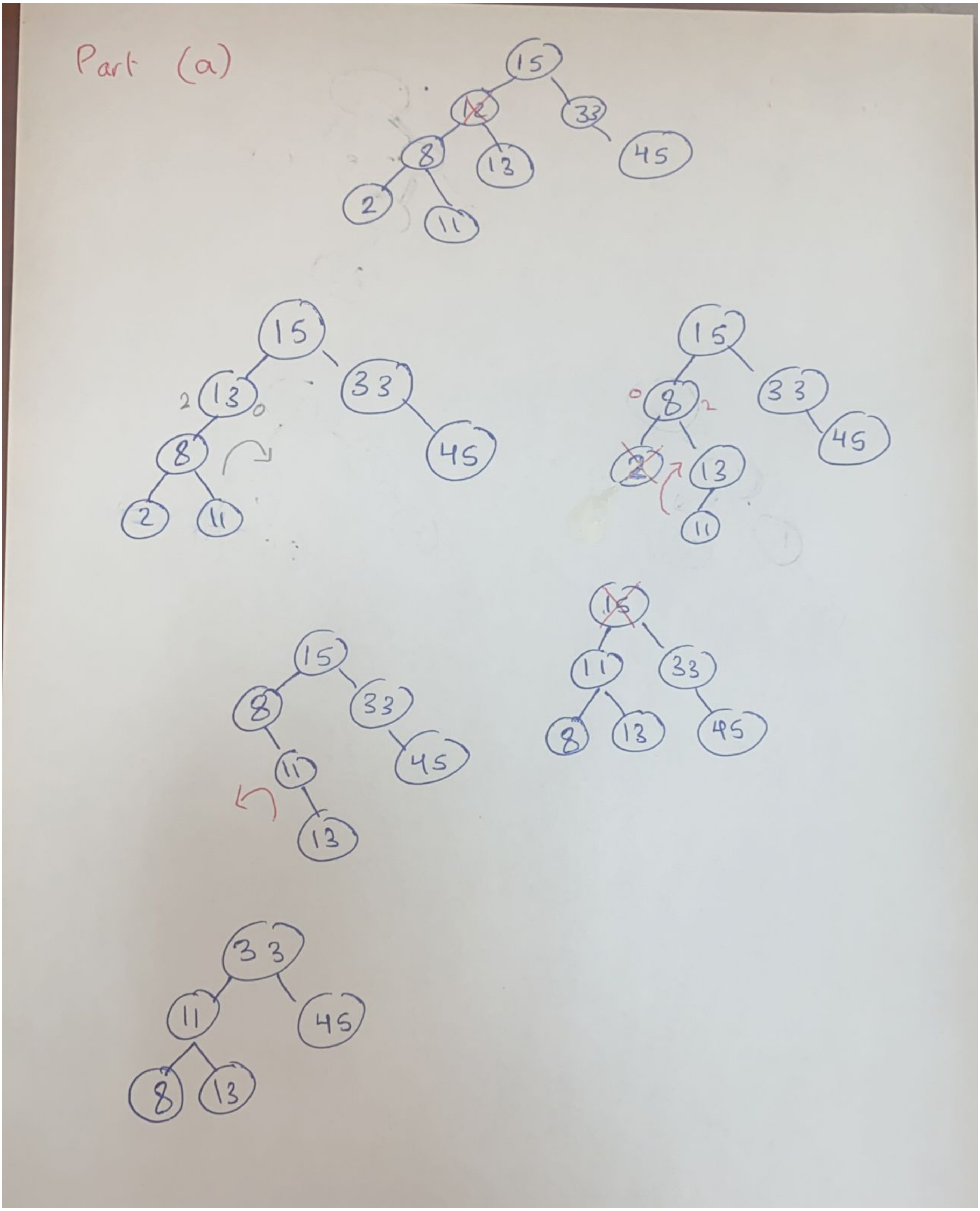
Question 3[/5 marks]

Part a [2 points]: Consider the following AVL Tree

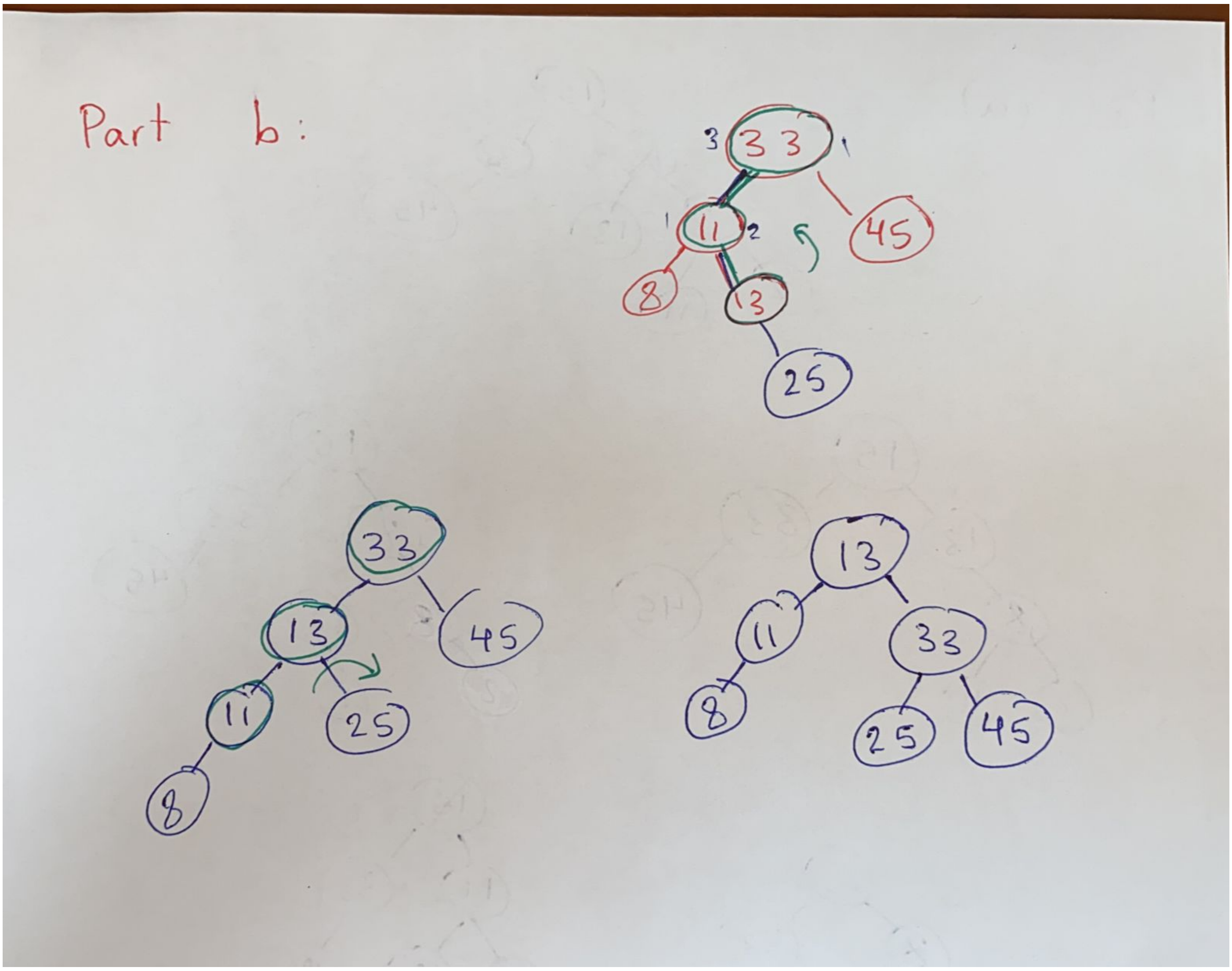


Remove these keys from the tree. For each removal, identify the deletion case AND show the appropriate rotations to balance this tree.
Note: Perform each of the removals on the resulting tree.

12, 2, 15.


Part (a)
15
12
33
8
13
45
2
11
15
13
33
8
45
2
11
2
0
15
8
33
2
13
45
11
0
2
15
8
33
11
45
13
15
11
33
8
13
45
33
11
45
8
13

Part b [1 point]: In the resulting tree from part a, insert **25**. Show appropriate rotations to balance your tree.



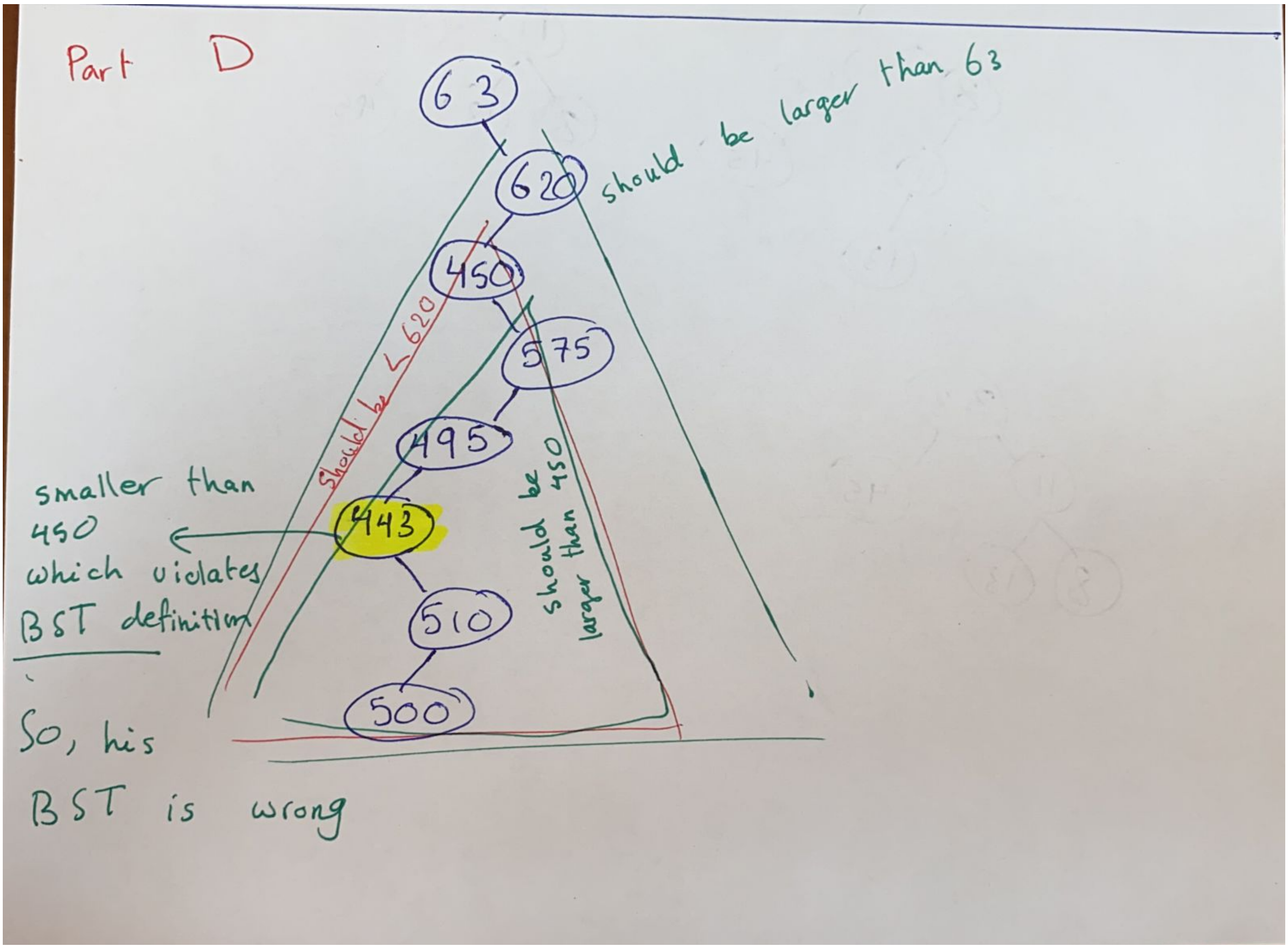
Part c [1 point]: In the resulting tree from part (b), do a Post-Order traversal. Show the sequence of the output generated.

8 11 25 45 33 13

Part d [1 point] Ahmed drew a BST of values ranging from 1 to 1000. Searching for value 500, Ahmed decided to print all nodes starting from the root until the search reaches the node containing value 500. The following sequence is printed on screen. Is Ahmed's BST Correct or Wrong? Justify your answer.

63, 620, 450, 575, 495, 443, 510, 500.

It is wrong, because the sequence which is printed violates the BST definition as 443 is smaller than 450 and it should be on the left while it comes on the right


Part D
63
620
should be larger than 63
450
575
495
443
510
500
should be < 620
should be larger than 450
smaller than 450 which violates BST definition
So, his BST is wrong