COMPUTER ENGINEERING DATA STRUCTURES MIDTERM

Name/Surname: _____

Student Number: _____

Problem 7 (15 points) Suppose that the Java library java.util.LinkedList is implemented using a doubly-linked list, maintaining a reference to the first and last node in the list, along with its size.

```
public class LinkedList {
    private Node first;
    // the first node in the linked list
    private Node last;
    // the last node in the linked list
    private int N;
    // number of items in the linked list
    private class Node {
        private Item item;
        // the item
        private Node next, prev;
        // the next and previous nodes
    }
}
```

Using the 64-bit memory cost model (int and pointers use 64 bit memory space), how much memory (in bytes) does a Node object use and how much does a LinkedList object use to store N items? Do not include the memory for the items themselves but do include the memory for the references to them.

- Memory of a Node:

8 + 8 + 8

- Memory of a LinkedList object:

8 + N(24) + 16

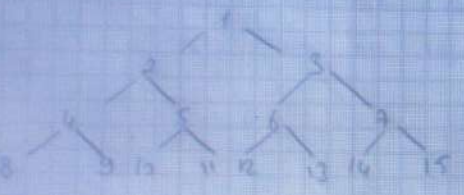
(a) What is the order of growth of the worst case running time of each of operations listed above (as a function of the number of items N currently in the data structure)?

2- Given a H-Heap storing 15[1 - 15] array-based representation of complete binary tree

a) Preorder → Root - Left - Right

b) Postorder → Left - Right - Root

c) Inorder → Left - Root - Right



Preorder = 1, 2, 4, 8, 9, 5, 10, 11, 3, 6, 12, 13, 7, 14, 15

Postorder = 8, 9, 4, 10, 11, 5, 2, 12, 13, 6, 14, 15, 7, 3, 1

Inorder = 8, 4, 9, 2, 10, 5, 11, 1, 12, 6, 13, 14, 7, 15, 3

13, 3, 14, 7, 15

3- [illegible] write a recursive algorithm that [illegible] in array A of integers containing of integer A[i] that is the multiplication

COMPUTER ENGINEERING DATA STRUCTURES MIDTERM

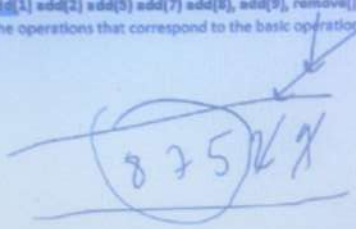
Name/Surname: _____

Student Number: _____

Problem 1 (10 points) Show the results of the following sequence of events, by drawing the state of the data structure: **add(1) add(2) add(5) add(7) add(8), add(9), remove(), remove()** Where add and remove are the operations that correspond to the basic operations in a:

a) Stack

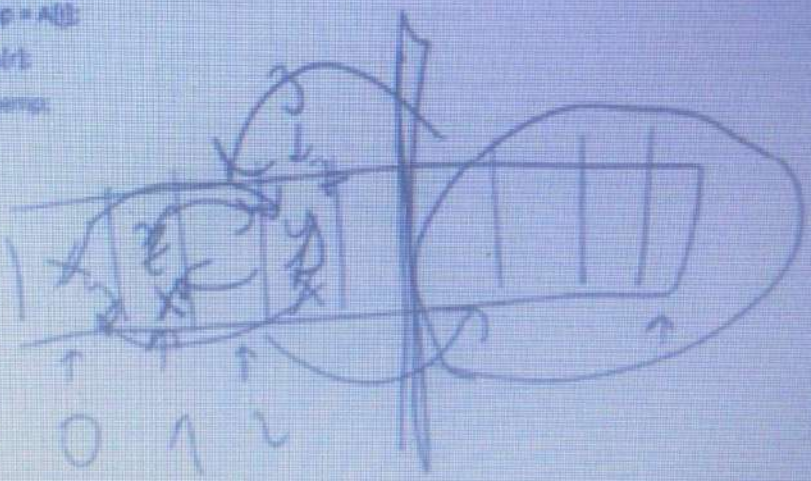
b) Queue

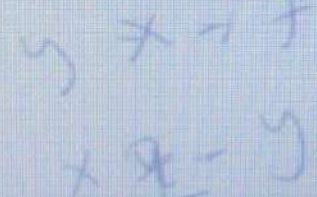


Problem 2 (15 points) For the input values inserted in the following order: 35 23 12 10 14 19 27 44 26 31 construct a heap.

Problem 8 - BONUS (15 points) Write a function, shuffle(A), that rearranges the elements of array A randomly. You may rely on the nextInt(n) method of the java.util.Random class, which returns a random number between 0 and n-1 inclusive.

```
shuffle(int[] A) {
    for(int i = 0; i < A.length; i++)
    {
        int r = Random.nextInt(A.length);
        int temp = A[i];
        A[i] = A[r];
        A[r] = temp;
    }
}
```





1- (20p) AVL TREE

Using the given sequence of inputs build an AVL tree, then remove the nodes in the given order from the tree.

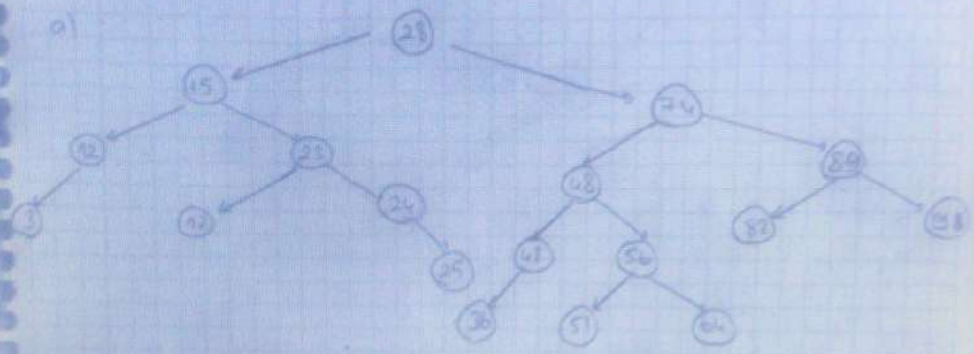
Input Sequence: 23, 64, 28, 12, 89, 15, 3, 98, 74, 82, 56, 19, 42, 66, 48, 24, 35, 51

Nodes to remove: 17, 89, 42, 56, 64

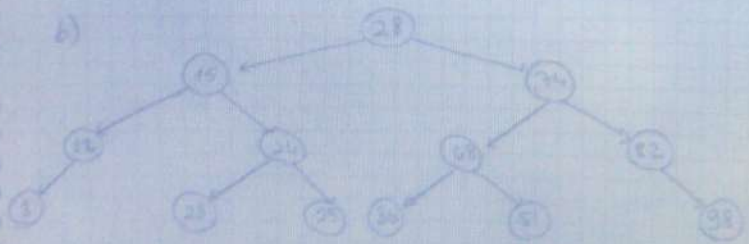
a) Show the tree after all the inputs are inserted.

b) Show the tree after all the nodes are removed.

a)



b)



Postorder = 8, 9, 4, 10, 11, 5, 2, 12, 13, 6, 14, 15, 7, 3, 1

Inorder = 8, 4, 9, 2, 10, 5, 11, 1, 12, 6, 13, 14, 7, 15, 3 ?

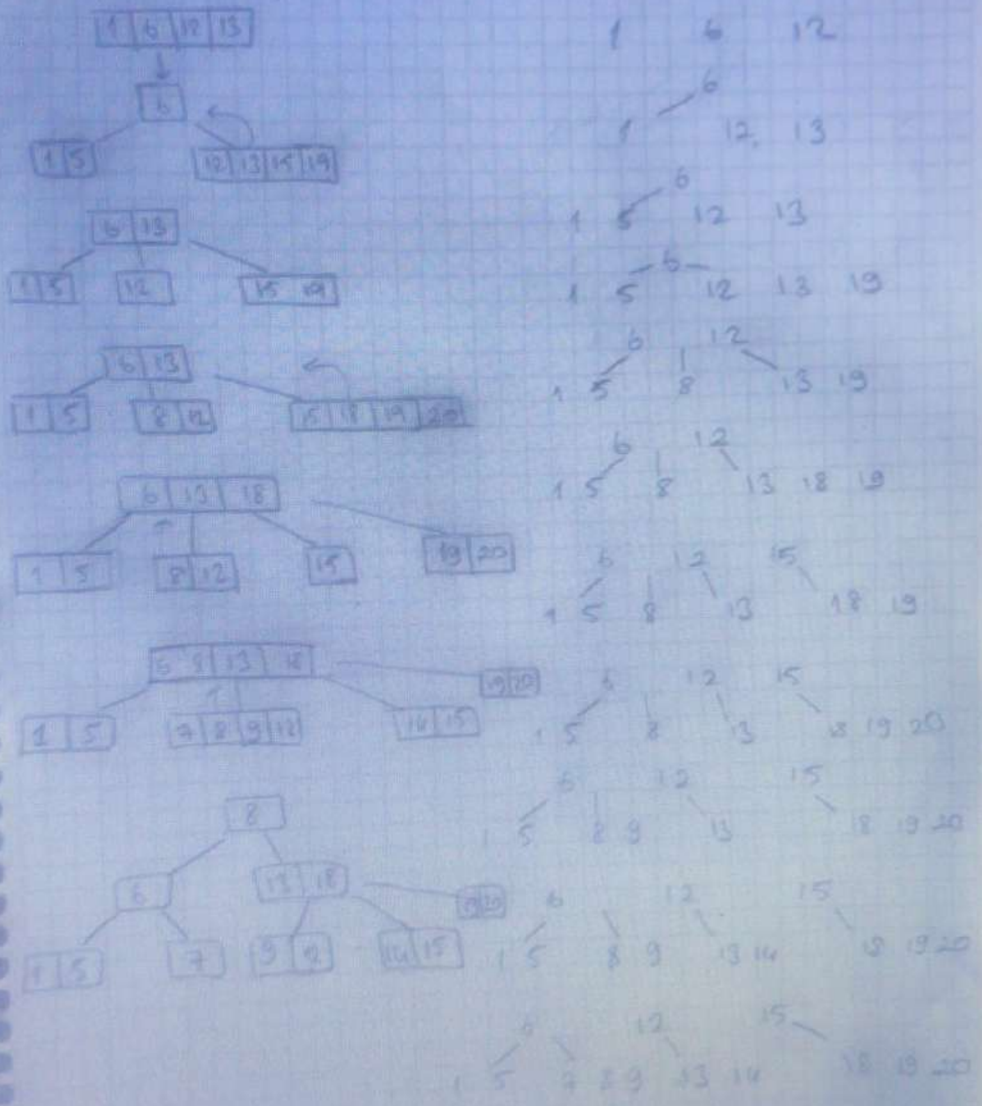
13, 3, 14, 7, 15

5- [20p] Write a recursive algorithm that will check if an array A of integers contains an integer A[i] that is the multiplication of two integers that appear earlier in A, that is, such that A[i] = A[j] * A[k] for j,k < i ?

```
int i=0;
int j,k=0;
boolean Recursion(int[] A, int i, int j, int k)
if(i>j && j>k){
if(A[i] = A[j] * A[k])
  return true;
else
  Recursion(A, i, j++, k) || Recursion(A, i, j, k++)
}
else
  Recursion(A, int, 0, 0)
else false;
```

7- Starting with an empty 2-4 tree insert each of the given letters into the tree in the order given

6, 12, 1, 13, 5, 19, 8, 18, 15, 20, 9, 14, 7

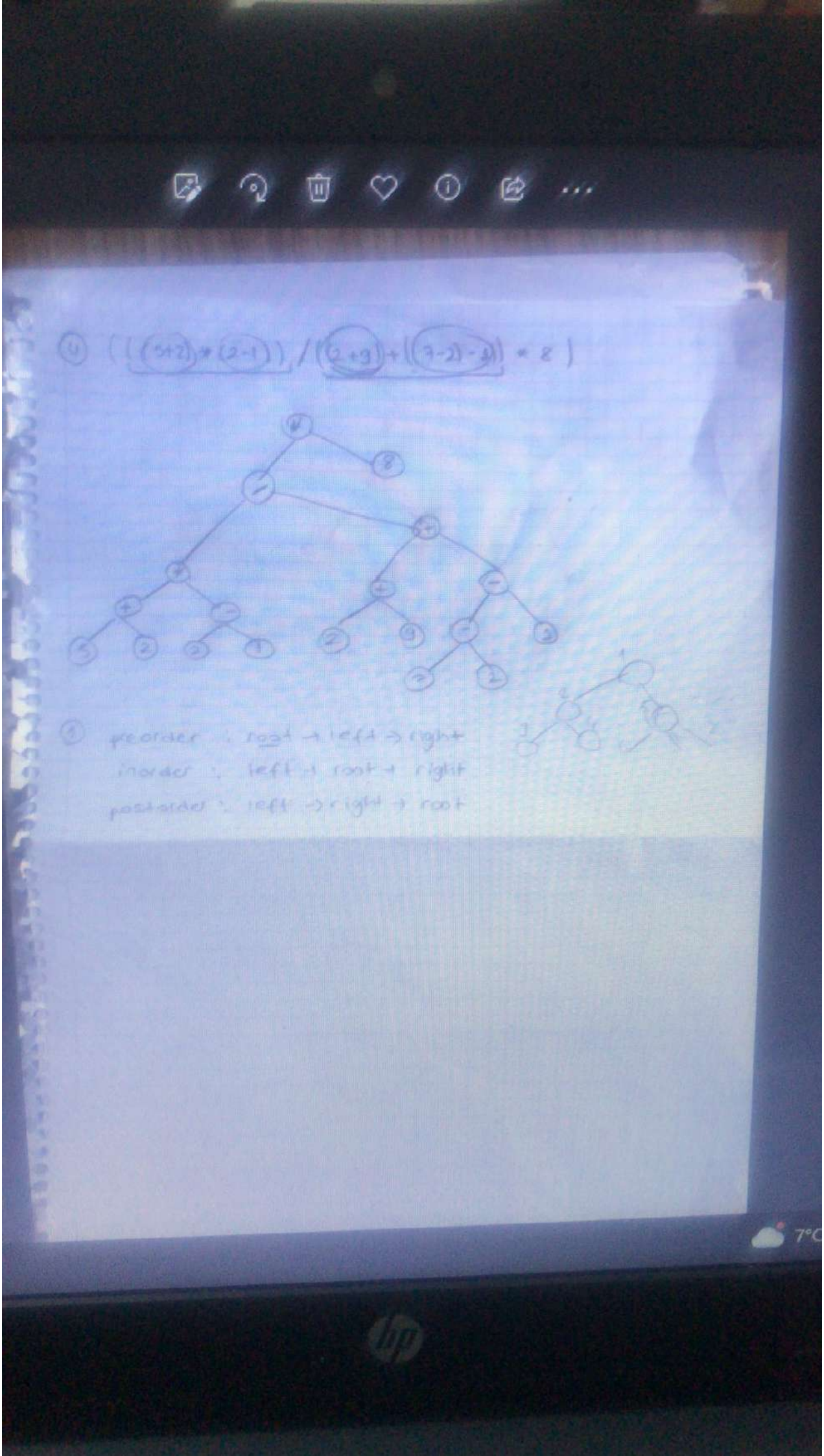


6.5 Give a recursive method for removing all the elements from a stack

```
void removeAllElements(stack<T> s) {
  if (s.isEmpty()) return;
  else {
    s.pop();
    removeAllElements(s);
  }
}
```

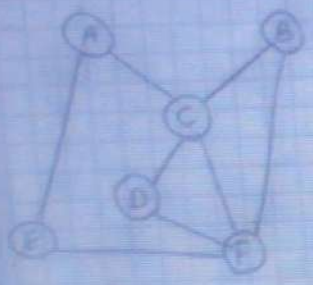
6.3 Initially, the stack is empty

Stack	Stack operation	Return value
()	push(5)	-
(5)	push(3)	-
(5,3)	pop()	3
(5)	push(2)	-
(5,2)	push(8)	-
(5,2,8)	pop()	8
(5,2)	pop()	2
(5)	push(9)	-
(5,9)	push(1)	-
(5,9,1)	pop()	1
(5,9)	push(7)	-
(5,9,7)	push(6)	-
(5,9,7,6)	pop()	6
(5,9,7)	pop()	7
(5,9)	push(4)	-
(5,9,4)	pop()	4
(5,9)	pop()	9
(5)		


preorder : root → left → right
inorder : left → root → right
postorder : left → right → root

5- a) (10 p) Show the adjacency matrix representation for the graph.

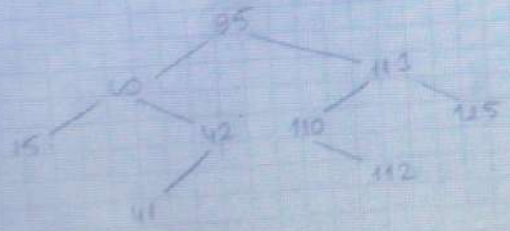
b) (5p) What would be the number of zeros in the adjacency matrix of the given graph

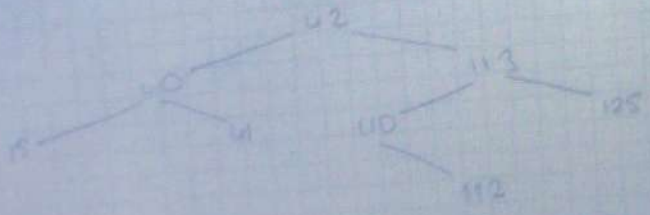


	A	B	C	D	E	F
A	0	0	1	0	1	0
B	0	0	1	0	0	1
C	1	1	0	1	0	1
D	0	0	1	0	0	1
E	1	0	0	0	0	1
F	0	1	1	1	1	0

b) 6x6 = 36
Total number of values in the matrix
There are 16 1's
So entries are 0's
36 - 16 = 20

6- (15p) Given the following Binary Search Tree, draw its value after deleting 95





4. Draw a representation of an initially empty list after performing the following sequence of operations

Add(0,4), Add(0,3), Add(0,2), Add(2,1), Add(4,5), Add(4,6), Add(5,7), Add(0,8)

Method	return value	contents
[0,4)	-	4
(0,3)	-	3,4
(0,2)	-	2,3,4
(2,1)	-	2,1,3,4
(4,5)	-	2,1,3,4,5
(4,6)	-	2,1,3,4,5,6
(5,7)	-	2,1,3,4,5,7,6
(0,8)	-	8,2,1,6,5,7,3,4

→ 0
4

→ 0 1
3 4

→ 0 1 2
2 3 4

→ 0 1 2 3
2 3 1 4

→ 0 1 2 3 4
2 5 3 1 4

→ 0 1 2 3 4 5
2 6 5 3 1 4

→ 0 1 2 3 4 5 6
2 6 5 7 3 1 4

→ 0 1 2 3 4 5 6 7
8 2 6 5 7 3 1 4

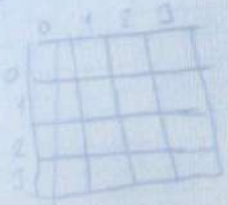
Date: Midterm-2019

1)

add(0,4), add(0,3), get(1), add(0,2), set(2,10), get(4), set(1,1), add(2,1), set(2,10), add(1,5), get(2), add(1,5), add(3,7), set(5,10), get(3), add(0,8)

Method	Return value	List contents
		4
add(0,4)	–	3,4
add(0,3)	–	3,4
get(1)	4	2,3,4
add(0,2)	–	2,3,10
set(2,0)	–	2,3,10
get(4)	null/error	2,3,10
set(3,14)	null	2,3,1,10
add(2,1)	–	2,3,10,10
set(2,10)	–	2,5,3,10,10
add(1,5)	–	2,5,3,10,10
get(3)	3	2,3,5,3,10,10
add(1,3)	–	2,3,5,7,3,10,10
add(3,7)	–	2,3,5,7,3,10,10
set(5,10)	–	2,3,5,7,7,10,10
get(3)	7	8,2,3,...
add(0,8)	–	

2) write a recursive algorithm to compute the sum of all elements in an nxn (two-dimensional) array of integers. What is your running time and space usage?




```
int find(A, i, j)
if (i, j) > A.length)
    return 0;
else
    return A[i][j]
find(A, int, j, total)
find(A, i, j+1, total)
```