12/10/17 MERGESORT(A) MERGESORT(A first half) MERGSORT(A 2nd half) MERGE INORDERTRAVERSAL(T) // T is a tree *if T = nil INORDERTRAVERSAL(T.left) then return output T.data INORDERTRAVERSAL(T.right) PRETRAV(T) output T.data PRETRAV(T.LEFT) PRETRAV(T.RIGHT) POSTTRAV(T) POSTTRAV(T.left) POSTTRAV(T.right) output(T.data) TRIANGLES(N) // N is an integer >= 0 if N = 0return 0 ¹⁰ from Notesale.co.uk Notesale.co.uk Notesale.co.uk Notesale.co.uk Notesale.co.uk Notesale.co.uk else N + TRIANGLES(N-1) Trace the algorithm with an input of N=6. Recursion Depth N N = 0 0 6 F 5 1 F 2 4 F 3 3 e 4 5 Т 6 0

output = 21

IB Exam Question: 6. Consider the following recursive algorithm FUN(X, N), where X and N are two integers.

```
FUN(X, n)
if N<=0 then
    return 1
else
    return X*FUN(X, N-1)
end if
```

The *return* statement gives the value that the algorithm generates.

- a) Determine how many times multiplication is performed when this algorithm is executed. [1]
- b) Determine the value of FUN(2,3), showing all of your working [3]
- c) State the purpose of this recursive algorithm [1]

Х	Ν	return
2	3	8
2	2	4
2	1	2
2	0	1

Value of FUN(2,3) = 8

a) N

c) It finds X^N

Applications

Торіс	Class Notes	Organised Notes	
5.1.18 Define the term dynamic data structure.		Dynamic data structures are data structures that can change size during the execution of a program. The size of the structure is determined during run time, which is a very efficient use of memory space.	
5.1.19 Compare the use of static and dynamic data structures.		Static data structures	Dynamic data structures
		Computer can allocate space during compilation Easy to program Easy to check for overflow An array allows random access	Only uses the space needed at any time Makes efficient use of memory Storage no longer required can be returned to the system for other use
		Programmer has to estimate maximum amount of space needed Can waste space	Difficult to program Can be slow to implement searches A linked list only allows serial search
5.1.20 Suggest a suitable structure for a given situation.		Stacks: The most important application of a stack is to implement function calls. This provides a technique for eliminating recursion from a program. Queues: Computing applications: serving requests of a single shared resource (printer, disk, CPU), Buffers: MP3 players and portable CD players, iPod playlist. Playlist for jukebox: add songs to the end, play from the front of the list. Handling interruptions, so the first interruption can be treated first Linked list: You need constant-time insertions/deletions from the list. You don't know how many items will be in the list. You want to be able to insert items in the midme of the lit. You want to be able to insert items in the midme of the lit.	
Pre,	view from Page	You have a set of the	ray ahead of time of memory for the elements in oplications where