

P#P###################################	
(a)	An algorithm is a way of solving a
	problem. It contains step(s) which
	are necessary to achieve a particular
V	task. Examples of represting an algorithm
4 \	are flowcharts and pseudocodes.
b)	Asymptotic analysis states that the
***************************************	exact memory and time required for an
***************************************	algorithm isn't important but rather
***************************************	we identify how they change with
	increasing problem size (ratio). Ex:
[In one platform on algorithm has O(n/2)
	complexity while in another platform it
	has O(1/4) complexity but ultimately
PP444444444444444444444444444444444444	both have O(n) complexity.
c)	$1. = O(n^3)$
	$ = o(n^2)$
4	iii = O(nlog(n))
	iv. = 0 (n)
	v. = o(n)/
d)	* The if statement in the function has
	O(1) complexity.
	+ The function will call itself (n-1)
***************************************	no of times.
	# final complexity = O(n-1) = { O(1)
7	= o(n).
>	
864866864898646864448644	
4444 h + 11 + 12 + 12 + 12 + 12 + 12 + 12 +	~



2. /0.	int bsea	rch (int	key, int	size	nt array[])		
di-			•	1			
، ط	1. // Initi	alize f	first to	0			
-H CHAD HOLIANINO DA RIA OÑ RA HOPA	_	lize 1	ast to	Siz volue	of size=1=		
3701771211A111 1-4 41624 112-77 F	3. // Initialize found to 0. A. // Initialize position to -1.						
а поминана дору 4 д франска поменя в							
8404600 huseban	6. // Set while loop condition.						
	7. 11 Calc						
	8.11 Set						
	9.11 Set	position	to midd	He if	keg is		
	foun	*					
	10, 11 Set	last t	o value o	of midd	le-1 if		
14	key	is les	ss than	arroy Tr	niddle].		
7	11, // Els						
44400 440444444444444444444444444444444		ddle +/1.					
-4 NB440 NB4N44 b4 NV44 NB4 B4 B4	12. // (10	se the	loop.	i da regenaroningo da gracie du nogene pe pro			
ha appead 1200 1220 1220 42 52 54 54 14 14	13. // Return the volue of position and terminate the function.						
**********************			***				
C.	Variable	Initially	Fleration 1	After Fleration 2	I teration 3		
	hey	23	23	23	23		
***************************************	Size	15	15	15/	15		
***************************************	Arst	Ø	8	8/	8		
***************************************	last	14	14	20	8		
	found	0	0	/ 0	0		
	position	-1	-1 /	-1	-1		
	C! found & &	true	true/	true	true		
* *************************************	first <= last)	₩A		24 hbahdawa101000*****************	***************************************		
***************************************	middle	NA	ブ	11	9		
**************************************	array [middle]	NA	9	30	25		
***************************************				×			
	}				·		



int min Index (floot dCJ, int size) {
int min In = 0 ; $i=1$;
float min = d (o);
海自for (; i < size ; i++)
2
if (dCiJ < min)
A min = d CiJ;
$/ \min In = i;$
/ >
return min In;
void swap (float *p1, float *p2) {
Float temps
temp = * p1;
$p_1 = p_2$
p1 = *p2; *p2 =/temp; }
void selection Sort (float d[], int size) {
F(size==1) return;
if (size >1) {
Swap (& d [min Index (d, size)], d);
return selection Sort (&d [1], size-1); }}
*
\$ ·



Ь,	* In the minIndex function, statements
	outside to for loop hore O(1) complexity.
	and inside the
	* Even though the for loop runs for decreasing
######################################	sizes, the complexity of it is still o(n).
<u> </u>	* Final complexity = $O(n) * O(1) + O(1)$.
	& The swap function has = O(n).
	* All statements in swop function have O(2)/
	complexity.
	\$ So, final complexity = O(1).
	<i>.</i>
	& In the selection Sort Function the Swap
	function has (0(2) complexity as
	already mentioned, but since it calls
	* Dough the function calls itself for
	decreasing sizes the min Index function,
	complexity is o(n).
	* The selection Sort function calls itself
	(n-1) of times though it is for decreasing
r =	sizes, ultimately complexity is O(n) here
	Thus final complexity is = Ofment
	$O(n) + O(n) \neq O(n^2)$

8448644	



4.	۵۰	Stack 3 5 st
L		ED
444744444		5 P
***************************************		b node.
<u> </u>	hingsingsing maps letters _ster	
		$\begin{array}{c ccccccccccccccccccccccccccccccccccc$
************	<u> </u>	data next data next data next
þýpgenennýði		
*********	b,	struct node makenode (float item) {
***************************************		struct made *p = (struct node *) malloc(sizeof)
		struct node * p = (struct node *) malloc (size of (struct node));
		if (!p) return 0; p -> data = item;
<i>[</i>		$p \rightarrow next = 0;$ return p;
~~ ##### 6 #A		
		void init (struct stack * s) {
	/	s -> sp = 0;
8,58,58,58,58,50		int full (struct stack * s) {
		return 1;
L		
المائدة		int empty (struct stack * s) {
		return $S \rightarrow SP = = 0;$
)	

84808497859		



сошил.	
***************************************	int push (struct stock s, float item) {
	struct node * p = makenode (item)
1 x 0 p m ² = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	if (!p) return 0;
· · · · · · · · · · · · · · · · · · ·	$if(empty(s)) s \rightarrow sp = P;$
	else
	$\langle p - \rangle \text{ next} = s - \rangle cp;$
ði þiádókkábunnurum ák séddföl	5->5p = p1/3
25\bbb	return 1;
PP-47-104-11114-14-14-14-14-14-14-14-14-14-14-14	
***************************************	float pop (struct stack *s) {
ester of the political of the update of the	45 Cempty (s))
	float temp = s -> sp -> data;
a easearo haebe docum ed pagaan	struct node * p = S -> sp;
	$S \rightarrow Sp = S \rightarrow Sp \rightarrow next;$
	free (p);
	return temp; }
के के के किसी सीकोर करते कु स्थान कु स्थानक साथ की हैं। ऐसी के के कु कु मू	float top (struct stock * s) {
	return S-> sp-> data; }

And sinds making because the description	
EA-minoracument to enterto entert	
" 1 2 1 2 5 m h mader à 1 4 2 4 2 2 4 4 4 2 4 2 4 4 4 1 1 1	
व EBN देवीक्ष्मान्त्रेष्ट्रक्षमान्त्रेष्ट्रक्षमान्त्रेष्ट	
المساورة والمساورة والمساو	

40-1-442	
P6248027862im = r== meq vaana pa	
788 Ad-yd-y-dygrygyn y 1146 084 86	
7 H 20000 Deld - 1900 dwga 422 4 62 54 24 2	

Write the number of the question in the column.	S S S S S S S S S S S S S S S S S S S
5. 8	6 (23)
######################################	(12) (45)

	(4) (3)(4) (2A) (56)

	(5) (6)
474544	

1884486-4	1 02 10 A 2 0 10 17 AC 0A 5/
b.	1. 23, 12, 4 , 3, 9, 13, 18, 45, 24, 56.
al Ball dell descript to the popular on popular	11. 3,4,9,12,13,15,23,24,45,56.
\	
	111. 3, 9, 4, 15, 13, 12, 24, 56, 45, 23.
a	(23)

,	(12) (AS)
14014114104144144	
1	(4) (3) (24) (56)
4	
	(3) (9) (5)

C	
***************************************	Struct node *left, *right;};
5/	
7	
原名 別点 だかわけ あわがり 見かみか 中 で でき 変 覧	



column.		
<u>d)</u>	void preorder (struct node tree.) {	
osanpunpapit-0404 bi apan mesan w	if (!tree) redurn 0	
Location delibration of the state of the s	print f ("/i" tree > dota);	
15	printf ("-/.i)t" tree -> data);	
	return preorder (tree -) left);	
	return preorder (tree -> right); }	
e)	struct node * find (struct node * tree, int key){	
	Struct node + temp;	
9444Maoran vana van an oun an o	if (!tree) return 0;	
William della della della special della supingua per mig py	if (tree -> data == key) return tree;	
	if (key < tree -> data) temp= find (tree -> left, key)5
1/	if (I temp) temp = find (tree -) right, key);	
Y	return temp; }	

# #\$4_4500m\\$\$4.5740m\0200000000		

P0788844884844444444		

-400141141114444444111		

A		

J9	
Write the number of the question in this column.	GREEN GREEN UNIVERSITY TOWN
,,,,,,,,	
ha pjete kod hi hi ha koh ki k he ede	
78 27 8 27 92 7 27 28 27 27 29 20 20 20 20	
1720421 10 10 11 10 1 10 10 10 10 10 10 10 10	
·	
há	
141	

~~~~	
~~~~	

háda	

Write the number of the question in this column.	GREEN GREEN UNIVERSUT TOWN
P400088844444444444	
- 4848 88 8448 8 4 4 5 - 4 2 8 PP	

1 > 9 9 17 11 9 11 9 11 9 1 1 9 1 9 1 9 1 9	{


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	
d dan han ann ann adar ar à ban ar a an am àr aid ann a'	
***************************************	
1	

Write the number of the question in this column.	GREEN UNIVERSITY TOWN
-11141417222271014717417717	
***************************************	
*************	
***************************************	
pel norm 2 d y 4 4 4 444 \$ \$4.5 \$ 6.6 \$ 10 10 10 10 10 10 10 10 10 10 10 10 10	
P E E P E E E E E E E E E E E E E E E E	
29299992222222222222222222222222222222	
***************************************	



# National School of Business Management BSc in Management Information Systems (UGC) -16.1

#### BSc in Software Engineering (UGC) -16.1

# BSc in Software Engineering /Computer Networks / Security (Ply) 16.1

# BSc in Computer Science – 16.1

## 1st Year 2nd Semester Examination Algorithms and Data structures - CS106.3

**Answer all Questions.** 

Time: 03Hrs

Date: 29th Mar 2017

#### Question 1 - 20 Marks

(a) Briefly explain what an algorithm is in the context of Computing.

[5 Marks]

(b) Briefly explain, giving an example, how asymptotic analysis can isolate the algorithm efficiency from the machine and platform dependency.

[5 Marks]

(c) Simplify the following Big-O expressions

[5 Marks]

i.  $O(2n^3+5n-10)$ 

ii.  $O(2n^2+10^2n)+O(n)$ 

iii.  $O(n)*O(\log(n))$ 

iv.  $O(n)+O(\log(n))$ 

v. n*O(1)

(d) Giving reasons, evaluate the time complexity of the following function.

[5 Marks]

int fact(int n)
{
 if(n==1) return 1;
 return n*fact(n-1);
}

### Question 2 - 20 Marks

Following code segment implements the binary search algorithm.

```
1
      first = 0:
      last = size -1; 15-1=14
2
3
      found = 0;
4
      position = -1;
                                            8+19=11 8+10=9
6
      while(!found && first <=last) {
7
      middle = (first+last)/2; (4 + 0 = 7)
8
      if(array[middle] == key) { found = 1;
9
                                               position = middle; }
             23 19 30 25
10
      else if(key<array[middle]) last = middle - 1;
11
      else first = middle + 1;
                                      9-2
12
13
      return position;
```

- (a) If the above code segment to write inside a function called bsearch() what will be the return type and required arguments for the function? Give your answer by writing the function header including return data type and argument declarations. [5 marks]
- (b) Write down a comment line you would include in the above code against each line to illustrate the function of each line or statement. You do not have to copy the code just put the line number and your comment in your answer script. [5marks]
- (c) Copy the following table into your answer script and complete it for each iteration for the problem scenario given below to carry out a desk-check of the code given above.

Variable	initially	After	After	After
		iteration 1	iteration 2	iteration 3
key	23			
size	15			
first	0			
last	14			
found	0			
position	-1			
(!found && first <=last)	true			
middle	NA			
array[middle]	NA			

array[	]													
2	5	9	10	12	15	18	19	23	25	29	30	35	43	45
0	1	2_	3	4	5	6	7	8	9	100	11	12	13	14
key =2	23						$\mathcal{T}$		1		P	[10 Marks]		
							1.				8-0			-

## Question 3 - 20 Marks

The following is a skeleton of a selection sort implementation in C.

```
int minIndex(float d[], int size){

// return the index of

// the min in the given array

void swap(float *p1, float* p2) {

// swap two vars

swap

return

return

// return the index of

// the min in the given array

// swap two vars

swap

return

return

// return the index of

// the min in the given array

// swap two vars

swap

return

return

return

// return the index of
```

- (a) Write C code to implement the above selection sort algorithm.
- [10 Marks]
- (b) Evaluate step by step, giving reasons, the time complexity of each of the above functions in terms of the Big-O notation. [10 Marks]

#### **Question 4 - 20 Marks**

Following code intends to implement a dynamic stack.

struct node{ float data:

```
struct node* next;
```

struct stack{ struct node* sp;

struct node* makenode(float item){ // make a new node with item

}

void init(struct stack * s){...} // initialize sp

int full(struct stack * s){...}

// return 1 if full

int empty(struct stack * s){...} // return 1 if empty

int push(struct stack *s, float item){ ...

float pop(struct stack *s){ ...

float top(struct stack *s){...}

- 7.0,6.2,20
- (a) Write a clear diagram to show the status of the stack structure instance, nodes, stored values and node linking after pushing the values 2.0, 6.2 and 7.0. [6 marks]
- (b) Write code for each function above to complete the stack implementation.

#### Question 5 - 20 Marks

(a) Draw a binary search tree generated by inserting the following items in the given order.

[4 Marks]

[14 Marks]

- (b) Draw the sequence of items you process, if the BST is traversed by,
  - pre-order, ( KULR)
  - in-order, ( ∠ v ₧)
  - post-order, (LRY) tree walking methods.

[6 marks]

(c) Write down a node structure in C, suitable to implement the above BST.

[2 marks]

- (d) Write a C function to display the above BST in pre-order traversal.
- [4 Marks]
- (e) Write a C function to find a value (key) in the BST by traversing the BST in pre-order manner.

[4 Marks]

*****End of the paper****