CITY UNIVERSITY OF HONG KONG

Course code & title	:	EE2331 Data Structures and Algorithms
Session	:	Semester B 2012/13
Time allowed	:	Two hours

This paper has SEVEN pages (including this cover page).

---

1. This paper consists of 9 questions in 2 sections.
2. Answer all questions in Section A and any two questions in Section B.

---

*This is a **closed-book** examination.*

No materials or aids are allowed during the whole examination. If any unauthorized materials or aids are found on a candidate during the examination, the candidate will be subject to disciplinary action.

Section A (50%): Attempt <u>all</u> questions from this section

Question 1: Consider the following two pieces of code:

```
void f(int a[], int b[], int n) {
  int i, j;
  for (i = 0; i < n; i++)
    for (j = 0; j < i; j++)
      if (a[i] < b[j]) a[i] = b[j];
}
```

```
int g(int a[], int n) {
  if (n <= 0) return a[0];
  return a[n]*g(a, n-1);
}
```

(a) What is the time complexity of function *f*? **(2 marks)**
(b) What is the space complexity of function *f*? **(2 marks)**
(c) What is the time complexity of function *g*? **(2 marks)**
(d) What is the space complexity of function *g*? **(2 marks)**

Question 2: Given that the stack has been implemented correctly, consider the following program:

```
void f(int key) {
  Stack s;
  stack_init(&s);
  push(&s, key);
  while (!stack_empty(&s)) {
    key = pop(&s);
    if (key >= 1) {
      printf("%d ", key);
      push(&s, key - 1);
      push(&s, key - 2);
    }
  }
}
void main() {
  f(4);
}
```

(a) What is the output of the above program? **(6 marks)**
(b) What is the minimum stack size required for the above program? **(1 mark)**

Question 3: Given the preorder sequence and inorder sequence of a binary tree, draw the corresponding binary tree.

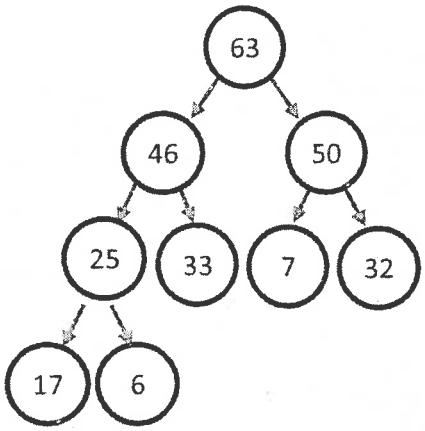
Preorder = *C H B G F E A D J I*
Inorder = *H B G C E F D A I J*

(8 Marks)

Question 4: Given the below max heap *h*, draw the result (please show the intermediate steps, if any) after:

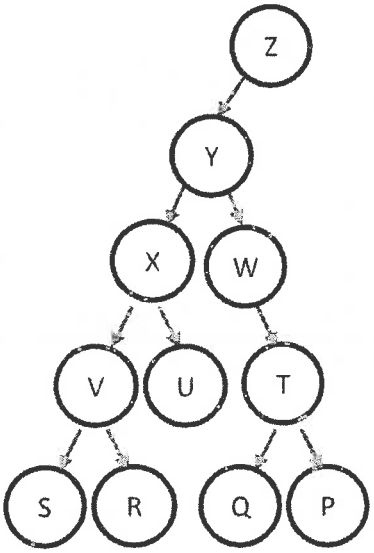
(a) Insert 49 into original *h*.
(b) Remove the max node from original *h*.

(6 Marks)



Question 5: A general tree can be represented by a binary tree. Given the below binary tree *g*, draw the original general tree.

(6 Marks)



Question 6: Given the following array *A*.

{6, 8, 5, 3, 4, 2, 1}

(a) The basic bubble sort algorithm is applied to sort the array *A* in ascending order. Show the array contents after each pass.

(4 Marks)

(b) The basic quicksort algorithm is applied to sort the array *A* in ascending order. Show the array contents after each pass.

(6 Marks)

(c) For bubble sort and quicksort, which of them is/are stable sorting algorithm(s)?

(1 Mark)

(d) Give one situation that:

(i) bubble sort performs better than quicksort

(ii) quicksort performs better than bubble sort

(4 Marks)

Section B (50%): Attempt any <u>two</u> questions from this section

Question 7: An integer queue is going to be implemented by a <u>doubly and linear linked list without dummy header</u>. The structure of linked list's node and queue is defined as follow:

```
typedef struct _node{
   int data;                  //value of the linked list node
   struct _node *prev;        //point to its preceding node
   struct _node *next;        //point to its succeeding node
} Node;                       //Node structure

typedef struct _queue{
   struct _node *front;       //point to the first element
   struct _node *rear;        //point to the last element
} Queue;                      //Queue structure
```

You may assume <stdlib.h> has been included. No linked list functions (e.g. create_node) nor queue functions (e.g. queue_empty) have been provided. You can call malloc() and free() to allocate and deallocate a block of memory respectively when necessary.

Complete the below functions:

void queue_destroy(Queue *q);
//to remove all nodes (if any) in the queue

(5 Marks)

void enqueue(Queue *q, int key);
//to insert a node into the end of queue

(10 Marks)

int dequeue(Queue *q);
//to return and remove the first node (if any) in the queue
//the function should report error if q is underflow

(10 Marks)

Question 8: The structure of the tree node of a binary tree is defined as follows:

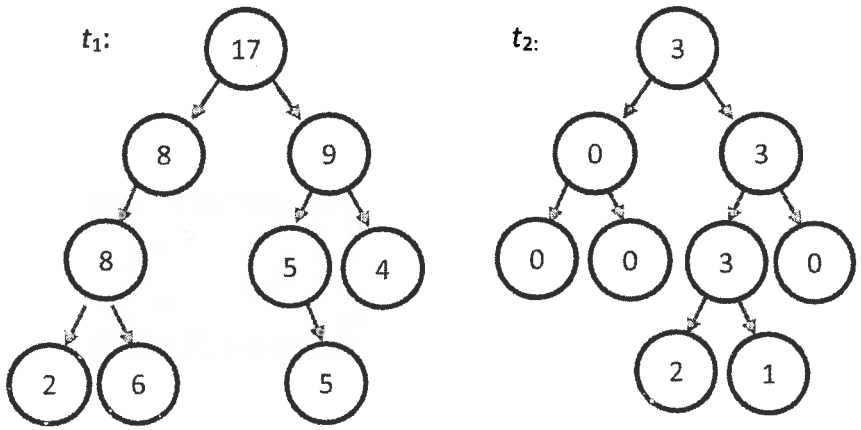
```
typedef struct _treenode {
   int data;                  //value of the tree node
   struct _treenode *left;    //point to its left subtree
   struct _treenode *right;   //point to its right subtree
} TreeNode;                   //Tree Node structure
```

(a) Given the root (*p*) of a linked binary tree, write a recursive function that prints the postorder sequence of the tree.

(5 Marks)

(b) Given an integer *k* and the root (*p*) of a linked binary tree, write a recursive function that returns the number of nodes on level *k*.

For example,
t_1 has 1 node, 2 nodes, 3 nodes and 3 nodes on level 0, 1, 2, 3 respectively.
t_2 has 1 node, 2 nodes, 4 nodes and 2 nodes on level 0, 1, 2, 3 respectively.



int count(TreeNode *p, int k);

(10 Marks)

(c) Given the root (*p*) of a linked binary tree, write a recursive function that tests if a binary tree has the following property:
the data in each non-leaf node equals the sum of the data in its children nodes.

For example, t_1 and t_2 are binary trees that satisfy the above property.

int test(TreeNode *p);
//return 1 if it satisfies the property, return 0 if it does not.

(10 Marks)

Question 9:

(a) The structure of linked list is defined as follow:

```
typedef struct _node {
   int data;                  //value of the linked list node
   struct _node *next;        //point to its succeeding node
} Node;                       //Node structure
```

(i) Given a linear linked list of integers without dummy header write a recursive function that prints the list in the following order (without modifying the list structure):

For example,

list1 = $\{a_1, a_2, a_3, a_4, a_5, a_6, ..., a_9, a_{10}\}$,
output = "$a_9\ a_{10}$ ... $a_5\ a_6\ a_3\ a_4\ a_1\ a_2$"

list2 = $\{a_1, a_2, a_3, a_4, a_5\}$,
output = "$a_5\ a_3\ a_4\ a_1\ a_2$" (note: the double quotations are not required)

(8 Marks)

(i) Given a linked list of integer without dummy header, write a non-recursive function that prints the list in above order (without modifying the list structure). You may assume the stack functions have been implemented and you can use the stack when necessary.

(10 Marks)

(b) Write a recursive function that binary searches a *key* from an ascending ordered array *a* with size *n*. The function should return the index of the *key* in array a. If *key* does not exist in the array, it should return -1.

(i) Write the recursive function.

(ii) Show the arguments of the initial recursive call.

(7 Marks)

- THE END -