Midterm exam 2014, questions and answers

Data Structures and Algorithms (Concordia University)

Part A [32 Points]

There are 8 multiple-choice questions in this part. To Choose an answer, simply draw a circle around the bullet (•) of the answer chosen (if any!). A correct choice for one question will get 4 point. An incorrect answer or marking several answers for a question will get **-1**. The minimum total mark for this part is 0.

1. Which of the following represents the best asymptotic ("Big-O") characterization of the function $f(n) = 3n \log n! + n^2 + 3$.

   • $O(n^n \log n)$ • $O(n^2 \log n)$ • $O(n! \log n!)$ • $O(n^2)$

2. Which of the following options list the function in **correct** non-decreasing order if they are compared by asymptotic growth?

   $n^3 \downarrow 9n^4$

   • $\log n, n, n \log n, (n^3 + \log n), (n^3 + 9\sqrt[2]{n^8}), n!, 2^n$

   • $\log n, n, n \log n, (n^3 + 9\sqrt[2]{n^8}), (n^3 + \log n), 2^n, n!$

   • $\log n, n, n \log n, (n^3 + \log n), (n^3 + 9\sqrt[2]{n^8}), 2^n, n!$

   • $\log n, n, n \log n, (n^3 + 9\sqrt[2]{n^8}), (n^3 + \log n), n!, 2^n$

3. An algorithm $A1(n)$ uses $2n - 2\sqrt{n}$ operations. Choose of the following pair of values for c and n_0 such that $c\,n \leq 2n - 2\sqrt{n}$ for $n \geq n_0$.

   • (1, 0) • (2, 1) • (1, 4) • (1, 1)

   $0 \leq 0$ $2 \leq 0$ $4 \leq 4$ $1 \geq 0$

4. If the array: [6, 21, 35, 3, 6, 2, 13] is added to a stack, in the order given above, which of the following is the top of the stack?

   • 2 • 6 • 13 • 35

5. Assume the sequence ADT based on a linked list implementation. Which of the following operations do not require $O(n)$ steps?

- addAfter
- add
- remove
- get

6. Consider this function declaration:

```
void quiz(int i)
{
      if (i > 1)
      {
         quiz(i / 2);
         quiz(i / 2);
      }
      cout << "*";
}
```

How many asterisks are printed by the function call quiz(5)?

- 3
- 4
- 7
- Some other number

7. Let the following circular queue can accommodate maximum six elements with the following data;

front = 2 rear = 4, Array:

		L	M	N	

What will happen after ***enqueue***(**O**) operation takes place?

- front = 3 rear = 5, Array:

		L	M	N	O

- front = 2 rear = 5, Array:

		L	M	N	O

- front = 3 rear = 2, Array:

O			L	M	N

- front = 2 rear = 4, Array:

	M	M	N	O	

8. Consider the following algorithm:

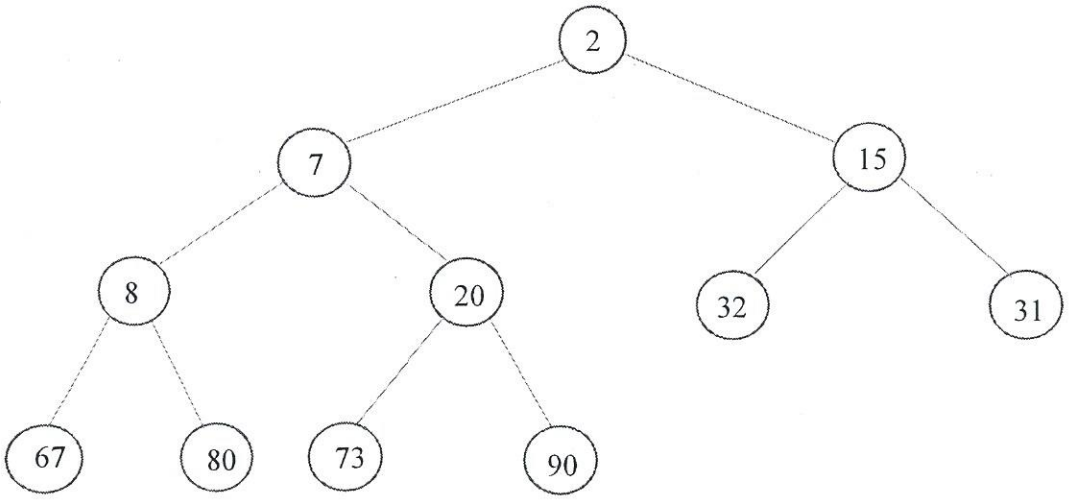
```
Algorithm T(r)
Input: Root r of a proper binary tree.

if r is a leaf then
     return 0
else
     p ← T(left child of r)
     q ← T(right child of r)

     if  p > q then
          return p + 1
     else
          return q + 1
```

What does the algorithm compute?

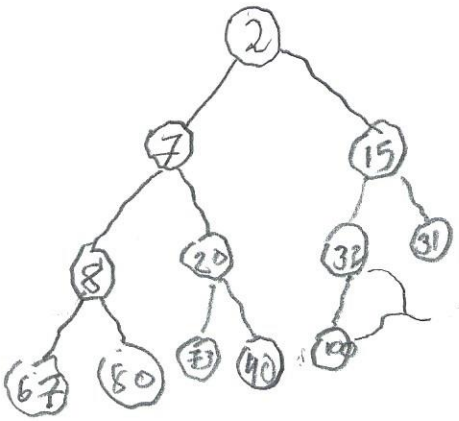
- The number of nodes in the tree.
- The number of internal nodes in the tree.
- The number of descendants of r.
- The height of the tree.

Part B [44 Points]

1. **[Heap-based Priority Queue – 24 Marks]**
   Consider the heap-based priority queue given below:

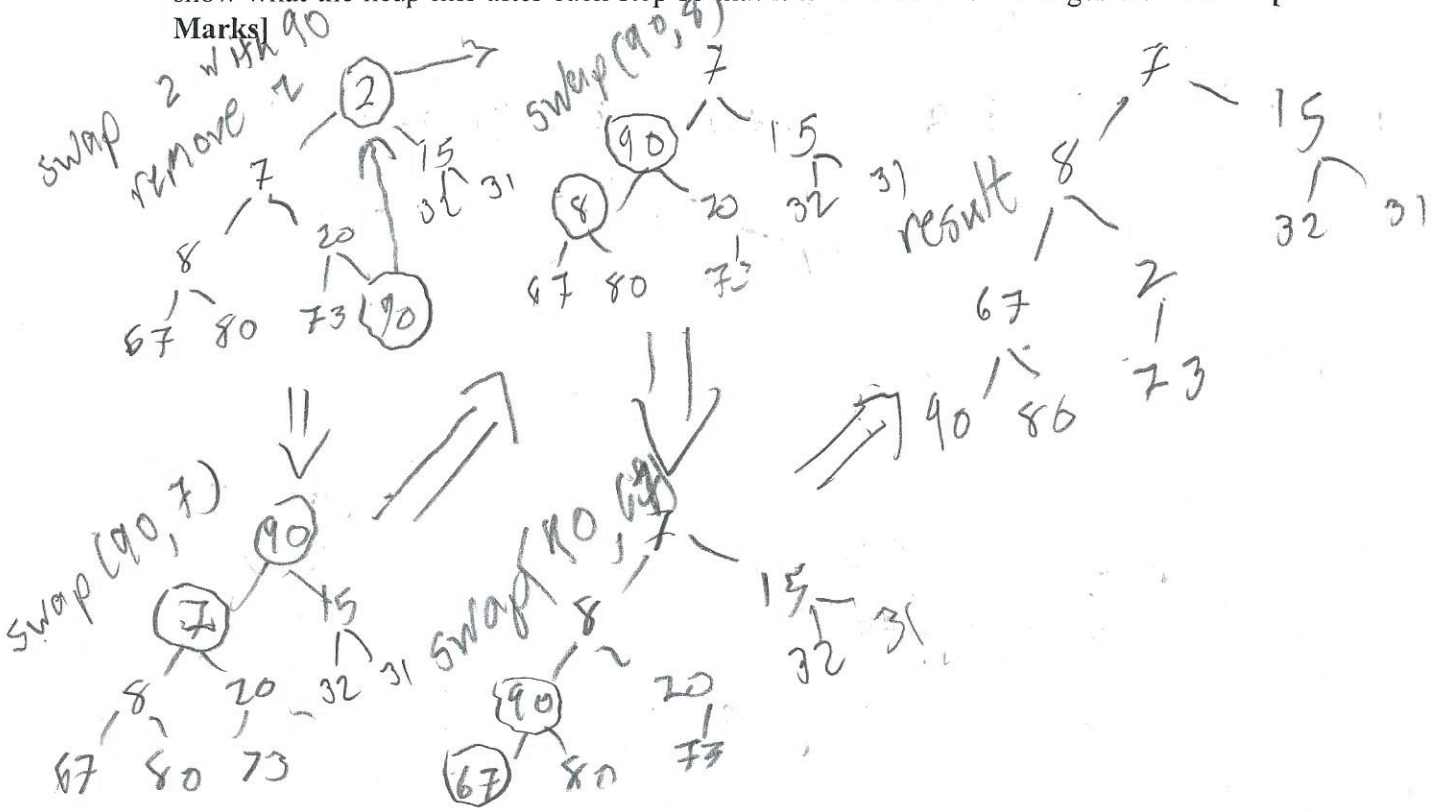


a) Show the series of steps that would take place in adding a node containing 100 to the above heap. In your answer, you should show what the heap like after each step so that it is clear how the changes were made. **[5 Marks]**



is parent > child = false no change

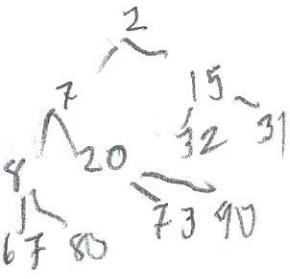
b) Consider again the heap shown in question 4 (before inserting 100). Show the series of steps that would take place when call **removeMin()** function. In your answer, you should show what the heap like after each step so that it is clear how the changes were made. **[5 Marks]**



c) If the heap is implemented using array, show the original heap in question 4 (before inserting 100) in an array. Be sure to show the array indexes. **[3 Marks]**

0	1	2	3	4	5	6	7	8	9	10
2	7	15	8	20	32	31	67	80	73	90

d) Consider again the heap shown in question 4 (before inserting 100). Write the output for pre-order, post-order and in-order traversal. **[3 Marks]**



Pre-order: 2, 7, 8, 67, 80, 20, 73, 90, 15, 32, 31

Post-order: 67, 80, 8, 73, 90, 20, 7, 32, 31, 15, 2

In-order: 67, 8, 80, 7, 73, 20, 90, 2, 32, 15, 31

e) Write a pseudo code for **pre-order** and **post-order** traversal algorithms. **[8 Marks]**

```
algorith pre-order (n)
input n node
   if n is Empty() then
      return
   print (value of n)
   pre-order( left child of n)
   pre-order( right child of n)
```

```
algorithm post_order(n)
input node n
   if n is empty then
      return
   post_order(left child n)
   post_order(righ child n)
   print (value of n)
```

DO NOT WRITE BEYOND THIS LINE

2.1 [Recursion- 10 Marks]

a) Write a recursive procedure for finding the maximum element of an array A containing n elements. (Note: your pseudo-code should not be more than a few lines) **[5 Marks]**

```
algorithm highest ( A, n, p)
input array A, length n, place p // n & p are int
  if n-1 = p then
     return A[p]
else
     m ← highest(A, n, p)
if m > A[p] then
     return m
else return A[p]
```

b) Calculate the time complexity for the above algorithm in terms of n. Show all necessary steps. **[3 Marks]**

will run n times // 2 ifs and 1 statement
max 4 statements will run for each if
thus equation 4n

2.2 [Recursion - 10 Marks]

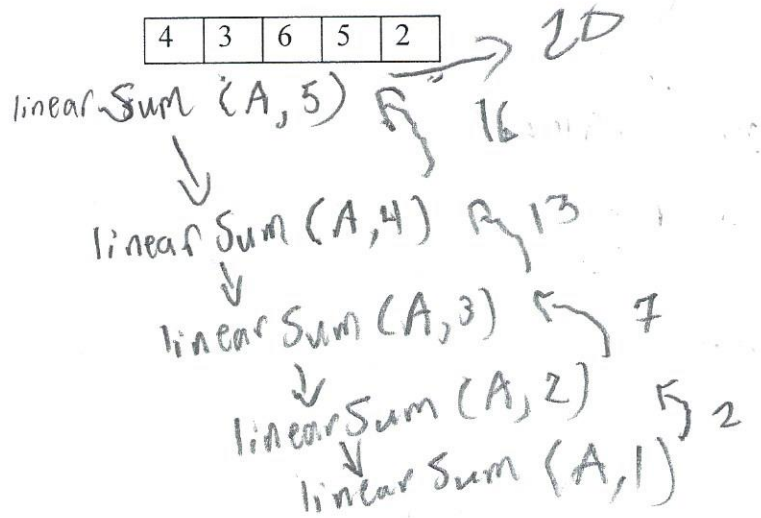
Consider the following pseudo code that computes the summation of all the element in an array.

```
Algorithm LinearSum(A, n):

Input:  An integer array A and an integer n >= 1, such that
        A has at least n elements
Output: The sum of the first n integers in A

if n = 1 then
     return A[0]
else
     return LinearSum(A, n - 1) + A[n - 1]
```

a) Write the recursion trace for executing the following array using LinearSum(A, n) function. **[4 marks]**

Assumed n = 5

4	3	6	5	2



b) Convert the above recursion algorithm as a binary recursion. [4 Marks]

```
LinearSum (A, n, p)
  input same as original with int p place
  if p > n-1
    return 0
  else
    return LinearSum (A, n, 2p) + LinearSum (A, n, 2p+1)
```

c) Can you convert the **LinearSum(A, n)** as tail-recursion? [2 Marks]

reverse last statements of final line

```
return A[n-1] + LinearSum (A, n)
```

Part C [24 Points]

In each of the following questions, please specify if the statement is **true** or **false**. If the statement is true, explain why it is true. If it is false, explain what is the correct answer is and why. (For each question, 2 points for the true/false answer and 4 points for the explanation). There is **no penalty** for selecting the wrong answer.

1. When you use a heap to implement a priority queue, the time complexity of inserting n elements into the heap is $O(\log n)$.

☒ True ☒ False

each insert is $O(1)$? n
and happens n times

-4

2. If $f(n) = n^2 + 5\,n \log n$, then $\Theta(n^2)$.

☑ True ☐ False

$n^2 + 5n\log n \geq n^2$

$n^2 \geq n^2$

$5n^2 \geq 5n\log n$

$n^2 + 5n^2$

$6n^2$

$c = 6$

3. If $f(n) = O(h(n))$ and $h(n) = O(k(n))$, then $f(n) + h(n) = O(k(n))$.

☑ True ☐ False

$f(n) \leq c_1 h(n) \leq c_2 k(n)$

4. Consider the following piece of code where n the variable data size is and K is a constant.

```
for (int i =1; i < n; i=i*K)
    for(int j=0; j< K; j++)
        sum[i] += j * sum[i];
```

The big-O complexity of the given code segment is $O(n^2)$.

☐ True ☑ False

first loop will run less than n times,
It will run $\frac{K}{n}$ times