Computer Science Foundation Exam

August 18, 2000
Solution for CS 1 sections

(1, 20%) Given the following array of numbers and algorithm, answer the questions below. Assume that the global array **X[1..n]** is correctly declared and contains the values shown.
Assume that the procedure was called with **`S(1, 6).`**

Array X	4	5	2	6	3	5
position	1	2	3	4	5	6

```
procedure S(i, j : integer)
  a, b, y, z : integer
  a ← 0
  b ← 0
  y ← 0
  z ← 0
  while (i < j) do
      if (X[i] < X[j]) then
          z ← z + j
          X[i] ← X[i] + i
          i ← i + 1
          y ← y + X[i]
      else
          y ← y + i
          X[j] ← X[j] + j
          z ← z + X[j]
          j ← j - 1
      endif
      if (a <= b) then
          a ← X[i]
      else
          b ← X[j]
      endif
  endwhile
endprocedure
```

a) Show the array **`X`** after the procedure has completed execution?

Array X	5	7	5	6	8	11
position	1	2	3	4	5	6

b) What value will the following variables contain after the **`while`** loop is finished?

a	6	b	6	y	17	z	33

(2, 14%) The following are Postfix expressions. All values are single decimal digits and the operations are addition "+", subtraction "–", multiplication "*" and division "/".
In each box below the Postfix expression, show ONLY the contents of the stack at the indicated point in the Postfix string (point A, B or C). Put the final answer in the blank.
If the Postfix string is invalid, carry the operations as far as possible and write "invalid" as the answer.

a) **7 2 1 3 + ^A^ * - 5 3 + ^B^ * 2 / 6 ^C^ * = <u>-24</u>**

A	B	C
4		
2	**8**	**6**
7	**-1**	**-4**

b) **3 5 - 8 2 ^A^ / 4 6 - 7 2 ^B^ - * ^C^ + * = <u>12</u>**

A	B	C
	2	
	7	
2	**-2**	**-10**
8	**4**	**4**
-2	**-2**	**-2**

Next to each Postfix expression, circle one answer to indicate if it is a valid Postfix string or not:
(no extra credit for providing the answer, if it is valid)

c) **7 5 3 - + 3 / * 4 2 +** **Invalid**

d) **4 3 - 2 3 * 5 8 - + 4 - * 2 +** **Valid**

(3, 20%) Answer each of the following "timing" questions concerning an algorithm of a particular order and a data set of a particular size. Assume that the run time is affected only by the size of the data set and not its composition.

a) For an **O(n^3)** algorithm, one data set with **n = 5** takes **250** seconds.

How long will it take for a data set with **n = 3**? **<u>54 seconds</u>**

$$\frac{5^3}{250} = \frac{3^3}{x} \; ; \; x = 27*250/125 = 54$$

b) For an **O($n\log_2 n$)** algorithm, one data set with **n = 8** takes **96** seconds.

If you used a different-sized data set and it took **32** seconds, how large must that data set be? **<u>n = 4</u>**

$$\frac{8 \log(8)}{96} = \frac{n \log(n)}{32} \; ; \; n \log(n) = 8; n = 4$$

c) For an **O(2^n)** algorithm, a friend tells you that it took **8** seconds to run on her data set. You run the same program, on the same machine, and your data set with **n = 7** takes **64** seconds.

What size was her data set? **<u>n = 4</u>**

$$\frac{2^n}{8} = \frac{2^7}{64} \; ; \; 2^n = 8*128/64; n = \log(16) = 4$$

Given the following pseudocode segment, answer the questions below for an arbitrary `n`:

```
x ← 0
for i ← 1 to (2*n) do
     for j ← 1 to (3*n) do
        x ← x + i
```

d) What is the Order of this pseudocode segment? **<u>O(n^2)</u>**

e) What will be the value of `x` when the `for` loops end? **<u>$6n^3 + 3n^2$</u>**

$$\sum_{i=1}^{2n}\sum_{j=1}^{3n} i = \sum_{i=1}^{2n} 3ni = 3n\sum_{i=1}^{2n} i = 3n(2n+1)(2n)/2 = 6n^3 + 3n^2$$

(4, 10%) In the space below, write a **recursive** algorithm called `PrintOdd`, that prints only the odd numbers from `1` to `n` in **increasing** order. The initial value of `n` may be either even or odd.

```
procedure PrintOdd(n)
     realPrint(1, n);
endprocedure;

procedure realPrint(current, n)
     if current <= n then
          print current;
          realPrint(current+2, n);
     endif;
endprocedure
```

There are, of course, many other correct solutions

--

(5, 18%) Find the closed form or exact value for the following:

(n is an arbitrary positive integer):

a) $\sum_{i=1}^{2n-1}(6i+5) =$ **12n² + 4n - 5**

$$= 6\sum_{i=1}^{2n-1} i + 5*(2n-1) = 6(2n)(2n-1)/2 + 10n - 5 = 12n^2 + 4n - 5$$

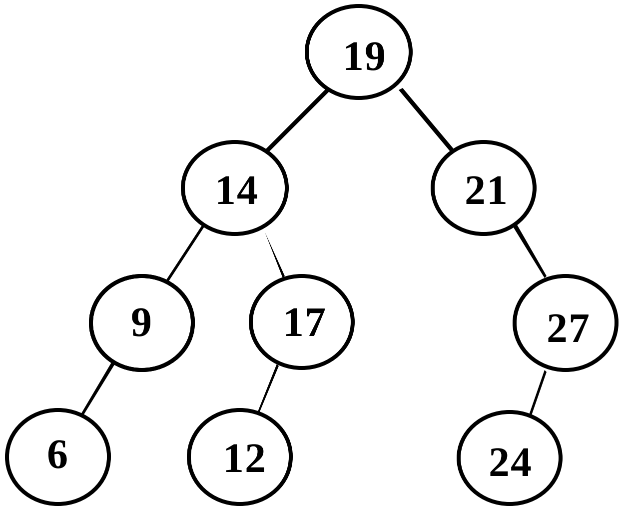
b) $\sum_{i=0}^{60}(2ni-4) =$ **3660n - 244**

$$\sum_{i=0}^{60}(2ni-4) = 2n\sum_{i=0}^{60} i - 4*61 = 2n(61)(60)/2 - 244 = 3660n - 244$$

c) $\sum_{i=40}^{100}(3i-6) =$ **12444**

$$\sum_{i=40}^{100}(3i-6) = 3\sum_{i=0}^{60}(i+40-2) = 3(61)(60)/2 + 3(61)(38) = (3)(61)(30+38) = 12444$$

(6, 18%) Given the following Binary Tree, answer the questions below :



a) Is this a valid Binary Search Tree? (circle one) **No**

b) List the nodes of this tree in the order that they are visited in a **postorder** traversal:

6 9 12 17 14 24 27 21 19

c) Perform the following procedure on the tree above, listing the output in the spaces below and leaving any unused spaces blank. Assume that the procedure is initially called with **`Problem_6(root, 22)`** and that the tree nodes and pointers are defined as:

```
tree_node definesa record
   data isoftype Num
   left, right isoftype ptr toa tree_node
endrecord
tree_ptr isoftype ptr toa tree_node
```

```
procedure Problem_6(node_ptr isoftype in tree_ptr,
                    key isoftype in Num)
   if (node_ptr <> NULL) then
      if (node_ptr^.data = key) then
         print(key)
      elseif (node_ptr^.data > key) then
         print(node_ptr^.data)
         Problem_6(node_ptr^.left, key)
      else
         print(node_ptr^.data)
         Problem_6(node_ptr^.right, key)
      endif
   endif
endprocedure
```

19 21 27 24