CSCE 2211 Spring 2024 Applied Data Structures
Midterm Exam (Closed Book)

Dr. Amr Goneid *(70 Minutes)* ***Date: Monday March 25, 2024***

Important:
In all questions, show in detail all steps leading to your answer. Points will be deducted if you do not show these steps in detail even if your answers are correct.

Question 1 (25 points)

- Convert the following infix expressions to postfix notation:
  (A+B)*((D+E)*C)+A **(10 points)**
  Answer: AB+CDE+**A+
- Given that: **A = 6, B = 2, C = 2, D = 1, E = 2**
  Evaluate the following postfix expression: **AB*CDE+*+C*** **(15 points)**
  Answer: 36

Question 2 (25 points)

A. Arrange the following in increasing order of complexity:

$$n^2 \log(3^n),\ \sqrt{n}\log(2^{\sqrt{n}}),\ (n+2)\log n^8$$ **(10 points)**

Solution:

$$n^2 \log(3^n) = n^3 \log 3 = O(n^3)$$

$$\sqrt{n}\log(2^{\sqrt{n}}) = n\log 2 = O(n)$$

$$(n+2)\log n^8 = 8(n+2)\log n = 8n\log n + 16\log n = O(n\log n)$$

$$\text{Then}\quad \sqrt{n}\log(2^{\sqrt{n}}) < (n+2)\log n^8 < n^2\log(3^n)$$

B. Consider the following algorithm to operate on an array ***a[1..n]*** of type double:

```
ALGORITHM Process (a [1..n], Smax, im, jm )
double Smax = 0.0;  int im =  jm = 0;  double Sij;
for i = 1 to n do
    Sij = 0.0;
    for j = i to n do
            Sij = Sij + a[j];
            if (Sij > Smax) { Smax = Sij;  im = i;   jm = j;}
```

(a) Find the exact number of double arithmetic operations done by the algorithm.
(b) What will be the complexity of the algorithm in terms of the array size (n)? **(15 points)**

Solution:

$$T(n) = \sum_{i=1}^{n}\sum_{j=i}^{n} 1 = \sum_{i=1}^{n}(n-i+1) = \sum_{k=1}^{n} k = n(n+1)/2$$

$$T(n) = O(n^2)$$

Question 3 (25 points)

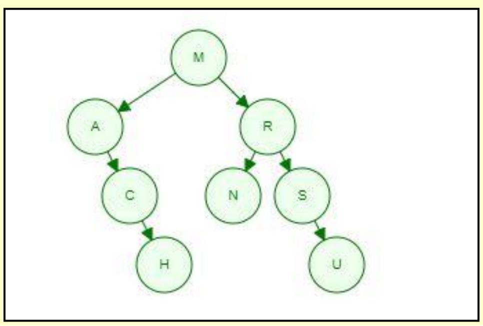
(a) In a complete binary tree, all levels are completely filled, except the rightmost of the last level. Given that the total number of nodes in the tree is N = 97. Find the height of this tree. **(10 points)**

Solution:

Let h be height of the tree. Since all levels from 1 to h-1 are completely filled, then the number of nodes in these levels is 2^{h-1}-1. If N is the total number of nodes in the tree, then:
2^{h-1}-1 < N ≤ 2^h-1. Adding 1 and taking logs, then **h-1 < log (N+1) ≤ h, or h-1 < log (98) ≤ h**
i.e **h-1 < 6.615) ≤ h. Therefore, h = 7.**

(b) Consider a **Binary Search Tree** containing characters. When traversed in post-order, it gives the following sequence: H, C, A, N, U, S, R, M. **Draw the tree.** **(16 points)**

Solution:



Question 4 (25 points)

The 7 character keys (A, B, C, D, E, F, G) are inserted in a BST using two different insertion orders: (G, F, E, D, C, B, A), and (C, B, A, E, D, G, F).

- Draw the two resulting trees. **(10 points)**
- Compute and compare the average search costs per key in the above two trees given that the search probability is the same for all keys. **(15 points)**

Solution:

- **The 1st tree is all left sided like a linked list.**
  The 2nd tree has the following level order traversal: Level(1): C, Level(2): B , E, Level(3): A, D, G, Level(4): F
- **Average Search Cost:**
  1st Tree: C = (1+2+3+4+5+6+7)/7 = 28/7 = 4 per key. Search Complexity is *O(n)*

2nd Tree: C = (1 + 2*2 + 3*3 + 1*4)/7 = 18/7 = 2.57 per key. Search Complexity is *O(log n)*
2nd Tree is more efficient for search because it is balanced.

Good Luck