CSCI 200

HW #1: Insertion sort, analyzing algorithms, growth of functions, basic data structures

Directions: *Complete your work on a separate sheet of paper. Submit the physical copy of your work at the beginning of class on the specified due date. Show your work. You may work in groups of up to 3 students provided that all students participate in each question; write all names at the top. Unless otherwise stated, assume that logarithmic functions are using base 2. Provide a short preliminary explanation of how an algorithm works before running an algorithm or presenting a formal algorithm description, and use examples or diagrams if they are needed to make your presentation clear.*

1. Using Figure 2.2 as a model, illustrate the operation of `INSERTION-SORT` on the array $A = [10, 42, 30, 28, 53, 42]$.

```
function Foo(a, n)
    k ← 0
    b ← 1
    while k < n do
        k ← k + 1
        b ← b * a
    return b
```

```
function Bar(a, n)
    k ← n
    b ← 1
    c ← a
    while k > 0 do
        if k mod 2 = 0 then
            k ← k/2
            c ← c * c
        else
            k ← k − 1
            b ← b * c
    return b
```

```
function Baz(n)
    k ← 1
    m ← 1
    while k ≤ 6 do
        m ← m * k
        k ← k + 1
    return m
```

```
function Moo(n)
    k ← 1
    x ← 1
    while k ≤ n do
        m ← 1
        while m ≤ n do
            x ← m * k
            m ← m + 1
        k ← k + 1
    return x
```

2. (a) Consider the algorithm `Foo`, which takes as input integers a and n. Describe what the algorithm `Foo` does. *Hint: 'It computes ______.' Fill in the blank with a mathematical expression.*

   (b) Analyze the worst-case running time of `Foo` and express it in O-notation. Justify.

3. (a) Consider the algorithm `Bar`, which takes as input integers a and n. Describe what the algorithm `Bar` does. *Hint: 'It computes ______.' Fill in the blank with a mathematical expression.*

   (b) Analyze the worst-case running time of `Bar` and express it in O-notation. Justify.

4. (a) Consider the algorithm `Baz`, which takes as input integer n. Describe what the algorithm `Baz` does. *Hint: 'It computes ______.' Fill in the blank with a mathematical expression.*

   (b) Analyze the worst-case running time of `Baz` and express it in O-notation. Justify.

5. (a) Consider the algorithm `Moo`, which takes as input integer n. Describe what the algorithm `Moo` does. *Hint: 'It computes ______.' Fill in the blank with a mathematical expression.*

   (b) Analyze the worst-case running time of `Moo` and express it in O-notation. Justify.

6. Express each of the following functions using O-notation. That is, find the slowest growing function $g(n)$ such that $f(n) \in O(g(n))$.

   (a) $f(n) = 10n + 32$

   (b) $f(n) = 5\log(n) + 42$

   (c) $f(n) = 18 + n^5 + 2^n$

   (d) $f(n) = 4 - 4n + 4n^2 + 4n^3$

   (e) $f(n) = 5\log(n) + 42n$

7. Rank the following functions by order of growth, from slowest-growing to fastest-growing. That is, find an arrangement $f_1, f_2, f_3, ..., f_{10}$ of the following functions satisfying $f_1 \in O(f_2)$, $f_2 \in O(f_3)$, etc.

$2^{\log_2(n)}$	n^{100}	100	$n^{0.001}$	$log_2(n)$
$\log_2^3(n)$	$n\log_2(n)$	2^n	n^2	$\sqrt{n}$

8. (a) You are given an initially empty stack S stored in array $S[1...10]$. Perform the following operations on it: PUSH(S, 'B'), PUSH(S, 'A'), PUSH(S, 'T'), PUSH(S, 'I'), POP(S), POP(S), PUSH(S, 'Z'), PUSH(S, 'A'), POP(S), PUSH(S, 'I'), PUSH(S, 'N'), PUSH(S, 'L'), POP(S), PUSH(S, 'G'), PUSH(S, 'A'), PUSH(S, 'R'), PUSH(S, 'F'), POP(S), POP(S). Show the contents of the stack after all operations have been performed, using figure 10.1 as a model, and indicate where the top of the stack is.

   (b) You are given an initially empty queue Q stored in array $Q[1...10]$. Perform the following operations on it: ENQUEUE(Q, 'B'), ENQUEUE(Q, 'A'), ENQUEUE(Q, 'T'), ENQUEUE(Q, 'I'), DEQUEUE(Q), DEQUEUE(Q), ENQUEUE(Q, 'Z'), ENQUEUE(Q, 'A'), DEQUEUE(Q), ENQUEUE(Q, 'I'), ENQUEUE(Q, 'N'), ENQUEUE(Q, 'L'), DEQUEUE(Q), ENQUEUE(Q, 'G'), ENQUEUE(Q, 'A'), ENQUEUE(Q, 'R'), ENQUEUE(Q, 'F'), DEQUEUE(Q), DEQUEUE(Q). Show the contents of the queue after all operations have been performed, using figure 10.2 as a model, and indicate where the head and tail of the queue are.

9. Explain how to implement two stacks in one array $A[1...n]$ in such a way that neither stack overflows unless the total number of elements in both stacks together is n. The PUSH and POP operations for both stacks should run in $O(1)$ time.

10. Give the **pseudocode** for a linear-time algorithm for reversing a queue Q. To access the queue, you are only allowed to use the methods of a queue abstract data type (ENQUEUE, DEQUEUE, QUEUE-EMPTY). *Hint: You may use an auxiliary data structure.*