CS 6901   Capstone Exam Systems Spring 2016: Choose any 2 of the 3 problems.

1) Design a fully simplified 3-bit mod 6 up counter with your choice of T, JK, or D flip-flops. The circuit increments at each clock pulse, going through the sequence

0, 1, 2, 3, 4, 5, 0, 1, 2, 3, … .

Show the circuit diagram.

2) Consider the semaphore solution for the producer/consumer problem with a buffer of *n* elements.

common variables:

```
  semaphore A = ? ;
  semaphore B = ? ;
  semaphore C = ? ;
  the buffer, initialized as an empty fifo queue
```

code for producers:

```
  while (true) {
    produce();
    wait(A);
    wait(C);
    putiteminbuffer();
    signal(C);
    signal(B);
  }
```

code for consumers:

```
  while (true) {
    wait(B);
    wait(C);
    takeitemfrombuffer();
    signal(C);
    signal(A);
    consume();
  }
```

5 initialization choices for the semaphores are listed below.
Which of the 5 choices yield a correct solution?
For each choice that leads to an incorrect solution, answer the following 2 questions:
  Can deadlock occur?
  Can there be errors in how items are consumed?

Choice #1: A = *n*, B = 0, C = 0
Choice #2: A = *n*, B = 0, C = 1
Choice #3: A = 0, B = *n*, C = 1
Choice #4: A = 0, B = 0, C = 1
Choice #5: A = *n*, B = 0, C = 2

No explanations are needed. Simple yes/no answers to each question are sufficient for each choice.

3) Consider a system with 3 resources (A, B, C). The Banker's Algorithm is used to allocate resources and it has the following SAFE state:

Available:

A	B	C
1	2	2

Process	Allocation			Max			Need		
	A	B	C	A	B	C	A	B	C
P0	2	1	1	2	4	4	0	3	3
P1	1	1	2	2	4	4	1	3	2
P2	3	2	1	6	6	1	3	4	0
P3	0	1	0	0	3	2	0	2	2

The following 2 questions are independent. That is, assume that both begin with the same system data listed above.
a) If P1 requests an additional unit of resource B, will it be allowed? Justify your answer.
b) If P1 requests an additional unit of resource A, will it be allowed? Justify your answer.

<u>CS 6901 Capstone Exam Data Structures Spring 2016</u>: Choose any 2 problems.

1. Write a boolean function that is given a binary tree and returns true if and only if the tree has an even number of nodes. An empty tree is considered to have an even number of nodes.
Notes:
The function should have just one argument, a pointer to the root.
No global variables may be used.
No additional functions may be defined. You may not count the number of nodes.

2. Given the following two sorted arrays of integers:
A[0]..A[n-1],
B[0]..B[m-1].
Write an algorithm (using code) that merges the contents of A and B into a new sorted array of integers
C[0..n+m-1].

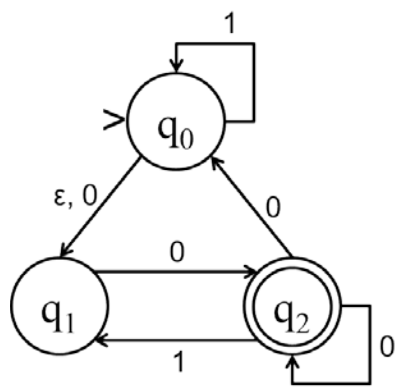
Your algorithm must run in $\mathbf{O}(n+m)$.

3. Let $G = (V, E, W)$ be a connected undirected weighted graph with $n = |V|$ vertices and $m = |E|$ edges. The edges are stored using the adjacency lists implementation.
a) Write a function that constructs a linked list of the edge weights in G in depth-first search order. Return a pointer to the first entry in the list.
b) State in big-theta terms the runtime of your routine as a function of n and/or m.
c) Suppose the graph had been given as an adjacency matrix. For this implementation, state in big-theta terms the runtime of your routine as a function of n and/or m.

Theory Exam

Answer **ANY TWO** of the following three questions:

1. Convert the NFA shown below to an equivalent DFA using the standard method:



2. Provide a context-free grammar that generates the following language over $\Sigma = \{0,1\}$:

$$\{w = w \text{ contains at least as many 0's as 1's and } |w| \geq 2\}$$

3. Answer **TRUE** or **FALSE** for each of the following statement to indicate whether the conclusion is **<u>always</u> true**. If you do not know the answer, do not guess. **<u>Scoring</u>**: +2 points for correct answer; 0 point for no answer; -1 point for wrong answer.
   a. If $A \leq_p B$ and $A \in$ NP-Complete then $B \in$ NP-Hard
   b. If $A \leq B$ and A is not decidable then B is not acceptable
   c. If $A \leq_p B$ and $B \in P$ then $A \in NP$.
   d. If $A \leq B$ and $B \in P$ then $A \in P$.
   e. If $A \leq_p B$ and B is decidable then $A \in NP$.
   f. If $A \leq B$ and $B \in$ NP-Complete then A is decidable.
   g. If $A \leq_p B$ and $B \in$ NP-Complete then $A \in$ NP-Complete
   h. If $A \leq \bar{A}$ and A is co-acceptable then A is decidable
   i. If $A \leq_p B$ and $A \in$ NP-Complete then $B \in$ NP-Complete
   j. If $A \leq B$ and B is not decidable, then A is not decidable.