CS 692 Capstone Exam – Systems Spring 2019

Directions: Choose only two out of the three problems listed below. If you attempt all three, only questions one and two will be graded. Please show all work.

1) **(20 pts Total) DeadLock**

   a. (8pts) List the four standard necessary conditions of Deadlock.

   b. (8pts) Let P be a set of processes and R is a set of resources. The set E holds request or assignment edges of a process and resource.

      The sets P, R, and E are as follows:

      P = {P1, P2, P3}
      R = {R1, R2, R3}
      E = {P1->R1, P2->R3, P3->R2, R1->P2, R2->P2, R2->P1, R3->P3}

      R1 has one instance. R2 has 2 instances. R3 has one instance.

      Draw the resource allocation graph.

   c. (4pts) Is there deadlock in this situation? Briefly explain your answer.

2) **(20 pts Total) Critical Sections**

   a. (6pts) List and define the three requirements of a solution to the critical section problem.

   b. (8pts) Consider the proposed solution of the critical section problem listed below. Does the proposed solution meet each of the three requirements? Explain why or why not.

```
Process A             Process B
while (true)           while (true)
{                     {
  wait(mutex);            wait(mutex);
  //critical section     //critical section
  signal(mutex);          signal(mutex);
}                     }
```

c. (6 pts) If the signal(mutex) call in process B is omitted? Will this change your answers from part b) above. Explain why or why not.

3. (20 pts Total) Scheduling

a. Below is a set of processes with CPU burst times listed in milliseconds.

Process	Arrival Time	CPU burst
1	0	10ms
2	5	5ms
3	2	2ms
4	6	8ms
5	1	12ms

a. (8 pts) Draw a Gantt chart for the ***Shortest Remaining Time First*** scheduling algorithm. Label the ending times of each process.

b. (8 pts) Draw a Gantt chart for the ***Round Robin*** scheduling algorithm. Assume round robin ordering starts with Process 1. Use a time quantum of ***4ms***. Label the ending times of each process.

c. (4 pts) For Round Robin in part b) above, what is the ***average wait time***? What is the ***turnaround time***?

CS 692   Capstone Exam   Algorithms   Spring 2019: Choose any 2 of the 3 problems.

1) Given a nonempty binary tree, write a function that returns the number of leaves in the tree.
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.

2) Solve the recurrence relation $T(n) = T(^{n}/_{2}) + log_2(n)$ where $T(1) = 1$ and $n = 2^k$ for a nonnegative integer *k*. Your answer should be a precise function of *n* in closed form. An asymptotic answer is not acceptable. Justify your solution.

3) Let *G* be a connected undirected graph with vertices v[0], … , v[n-1] and *m* edges. The edges are stored using the adjacency lists implementation.
a) Write code that prints the vertex numbers in breadth-first search order.
For any local data structure you use (such as stacks, queues, etc.), you may assume that the basic operations already exist (and so you don’t need to write the code for pop, enqueue, etc.).
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*.

CS 692 Capstone Exam Theory Spring 2019

Choose only two out of the three problems listed below. If you attempt all three, only questions one and two will be graded. Please show all work.

1). Consider $\Sigma = \{a, b\}$:

- **a.** (5 points) State the Pumping Lemma for regular languages.
- **b.** (15 points) Prove whether or not the following language is a regular language:

$$L = \{ a^i b^j : i > j > 0\}$$

2). Consider $\Sigma = \{a, b\}$:

- **a.** (5 points) Give one example of a context-free language.
- **b.** (5 points) Give one example of a language that is not context-free.
- **c.** (10 points) Is the class of context free languages closed under intersection? Prove your answer.

3). The PARTITION Problem takes as input a set S of integers. The question is whether S can be divided into two subsets A and $\bar{A}$ such that $\Sigma_{x \in A}\, x = \Sigma_{y \in \bar{A}}\, y$?

- **a.** (5 points) How do you prove, in general, that a Problem X is NP-complete? Please give the steps.
- **b.** (15 points) Prove that the PARTITION Problem is NP-complete. You may assume that CNF-SAT, 3-CNF-SAT, VERTEX-COVER, HAMILTONIAN-CIRCUIT, and SUBSET-SUM are all known to be NP-complete.