CS 6901 Capstone Exam (Systems) Winter 2018: Choose any 2 of the 3 problems.

Question 1 - A page replacement policy is used to manage system resources. Suppose that a newly created process has 3 page frames allocated to it, and then generates the page references – ABCBADABCDABACBD. For each of the following page replacement algorithms, show which references will cause page faults and show the contents of the 3 frames at the time of each fault. Assume that the frames are initially empty. You do not need to show the first 3 faults that are caused by demand paging.

(9 points) Least Recently Used (LRU)
(9 points) Second Chance
(2 points) Which of the above page replacement algorithms suffers from Belady's anomaly?

Question 2 - Suppose the following jobs arrive as indicated for scheduling and execution on a single CPU. Assume that no time is lost during context switching.

Job	Arrival time	Time (msec)	Priority
A	0	12	1 (Gold)
B	2	4	3 (Bronze)
C	5	2	1 (Gold)
D	8	10	3 (Bronze)
E	10	6	2 (Silver)

Example Gantt chart:

	Job #	Job #	Job #
Time:	0	1	2

Draw a Gantt chart showing the scheduling for these jobs using the following algorithms.

(9 points) Round Robin (RR) (quantum = 4) – (assume all jobs have already arrived)
(9 points) (Preemptive) PRIORITY
(2 points) What is the average wait time for the (Preemptive) PRIORITY scheduler?

Question 3 - (20 points) Draw a 1-bit comparator using only inverters and 2-input NAND gates. This comparator has 2, 1-bit inputs. Its output is 1 if and only if both inputs are the same. The input data lines of the comparator should be labeled starting with "I" (1:0) (I_1, I_0) and the output with "OUT."

CS 6901 Capstone Exam Data Structures and Algorithms Winter 2018

Choose any 2 of the 3 problems.

1.) Write the recursive function (in C/C++):

```
int CountKey(treeptr *p, int keyval);
```

that is given a (possibly empty) *binary tree* and returns the number of times a particular integer key (keyval) occurs *as a leaf node*. Include the declaration of any data structures. Do not write a complete program. Only the function is requested.

2.) Write the recursive function (in C/C++):

```
void quickSort(int arr[], int left, int right);
```

that implements the *quicksort* algorithm on an array of integers. Take the value of the middle element as the pivot value. Include the partition functionality in this same function. Use recursion to quicksort the array. Do not write a complete program. Only the function is requested.

3.) Solve the following recurrence relations (10 points each):

a. $T(n) = \begin{cases} 2T(n-1)+1, & n \geq 0 \\ 1, & n = 0 \end{cases}$

b. $T(n) = \begin{cases} T(n/3)+n, & n > 1 \\ 1, & n = 1 \end{cases}$

Theory Exam

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

1. Using the Pumping Lemma, prove that the following language over $\Sigma = \{0, 1\}$ is not context-free:

   $$\{0^n 1^n 0^{2n} : n \geq 0\}$$

2. Consider the following two languages:

   $\mathbf{A_{TM}} = \{M, w : M \text{ is a Turing machine that accepts string } w\}$
   $\mathbf{ALL_{TM}} = \{M : M \text{ is a Turing machine and } L(M) = \Sigma^*\}$

   Show that $\mathbf{A_{TM} \leq ALL_{TM}}$.

3. Answer each of the following questions with <u>only</u> **YES** or **NO** to indicate whether or not the first language class is a subset of the second. (Note: $A \subseteq B$ means that either $A = B$ or $A \subset B$.)
   Do not guess if unsure, as wrong answers will lower your score!
   <u>Scoring:</u> +2 points for correct answers; 0 points for no answers; -1 point for wrong answer

   a. regular $\subseteq$ co-acceptable?
   b. CFL $\subseteq$ NP?
   c. CFL $\subseteq$ acceptable?
   d. acceptable $\subseteq$ NP-hard?
   e. co-acceptable $\subseteq$ decidable?
   f. decidable $\subseteq$ NP-Complete?
   g. P $\subseteq$ NP?
   h. P $\subseteq$ NP-Complete?
   i. NP $\subseteq$ NP-hard?
   j. NP-Complete $\subseteq$ NP-hard?