

CS 6901 Capstone Exam Systems Spring 2015: Choose any 2 of the 3 problems.

- 1) a) Show how to construct a NOT gate from a single 2-input NAND gate.
b) Show how to construct a 4-input NAND gate using only 2-input NAND gates.
- 2) Consider the following two attempted solutions to the 2-process mutual exclusion problem. For each attempt,
 - a) Does the code guarantee mutual exclusion? If not, give an execution sequence where mutual exclusion is violated.
 - b) Could deadlock occur? If yes, give an execution sequence that leads to deadlock.
 - c) Is indefinite postponement possible? If yes, give an execution sequence that results in indefinite postponement.

Attempt #1: common boolean variables: flag1, flag2 (both initially false)

<pre>Process 1 while (true) { while (flag2); //empty body flag1 = true; Critical section; flag1 = false; Noncritical section; }</pre>	<pre>Process 2 while (true) { while (flag1); //empty body flag2 = true; Critical section; flag2 = false; Noncritical section; }</pre>
---	---

Attempt #2: common semaphore variable: S (initially 1)

<pre>Process 1 while (true) { wait(S); Critical section; signal(S); Noncritical section; }</pre>	<pre>Process 2 while (true) { wait(S); Critical section; signal(S); Noncritical section; }</pre>
--	--

- 3) Consider two CPU scheduling algorithms for a single CPU: Preemptive Shortest-Job-First (also known as Shortest Remaining Time First) and Round-Robin. Assume that no time is lost during context switching. Given five processes with arrival times and expected CPU time as listed below, draw a Gantt chart to show when each process executes using
 - a) Preemptive Shortest-Job-First (Shortest Remaining Time First).
 - b) Round-Robin with a time quantum of 5.

Of course, assume that the expected time turns out to be the actual time.

Process	Arrival Time	Expected CPU Time
P1	0	14
P2	2	9
P3	4	10
P4	8	4
P5	17	7

CS 6901 Capstone Exam Data Structures and Algorithms Spring 2015

Choose any 2 of the 3 problems.

1) Write a recursive function “CountNodes” of a (possibly empty) binary tree, which returns the total number of nodes in the tree. Include the declaration of your data structure.

2) Apply the Heap Sort algorithm to the following list. Draw the array after each step.
a[0] .. a[7]: {60, 50, 30, 10, 80, 70, 20, 40}

3) For each function with input argument n , determine the asymptotic number of “fundamental operations” that will be executed. Note that `fd` is recursive. Choose each answer from among the following. You do not need to explain your choices.

$\theta(1)$ $\theta(\log n)$ $\theta(n)$ $\theta(n \log n)$ $\theta(n^2)$ $\theta(n^2 \log n)$ $\theta(n^3)$ $\theta(2^n)$ $\theta(n!)$

a)

```
void fa(int n) {  
    for(i = 1; i <= n; i = i+2)  
        Perform 1 fundamental operation;  
    //endfor i  
}
```

b)

```
void fb(int n) {  
    for(i = 1; i <= n; i = 2*i)  
        Perform 1 fundamental operation;  
    //endfor i  
}
```

c)

```
void fc(int n) {  
    for(i = 1; i < n; i++) {  
        Perform 2 fundamental operations;  
        for(j = i; j <= n; j++)  
            Perform 1 fundamental operation;  
        //endfor j  
        for (k = 1; k <= n; k++)  
            Perform 1 fundamental operation;  
        //endfor k  
    } //endfor i  
}
```

d)

```
void fd(int n) {  
    if (n > 1) {  
        fd(n/2);  
        fd(n/2);  
        Perform n fundamental operations;  
    } //endif  
}
```

Choose any 2 of the 3 problems.

1). Give the state diagram for a DFA that recognizes the language:

$$L = \{w: w \text{ has prefix } 01 \text{ and suffix } 10\}$$

2). Show that the collection of decidable languages is closed under the operation of

- a. Concatenation.
- b. Kleene closure.

3). Answer **TRUE** or **FALSE** for each of the following statement to indicate whether the conclusion is **always true**. If you do not know the answer, do not guess.

Scoring: +2 points for correct answer; 0 point for no answer; -1 point for wrong answer.

- a. If $A \leq B$ and B is not decidable, then A is not decidable.
- b. If $A \leq B$ and B is decidable, then A is decidable.
- c. If $A \leq B$ and B is Turing recognizable, then A is Turing recognizable.
- d. If $A \leq B$ and B is not Turing recognizable, then A is not Turing recognizable.
- e. If $A \leq B$ and B is a regular language, then A is a regular language.
- f. If $A \leq B$ and $B \leq C$, then $A \leq C$.
- g. If A is Turing-recognizable, and $A \leq \bar{A}$, then A is decidable.
- h. If $A \leq_p B$ and $B \in \text{NP}$, then $A \in \text{NP}$.
- i. If a problem cannot be solved in polynomial-time, then it is NP-complete.
- j. If $A \leq_p B$ and B is NP-Complete, then A is in NP.