Identification Page

Name: _____

1. Write your name on this page, the Identification Page.
2. Do not include any identifying information on any other page of the exam (other than the existing number).
3. Verify that each page of the exam has the same number in the upper right-hand corner.
4. Do not attach the identification page of the exam to the other pages in any way.
5. Complete each problem.
6. You may use the front and back of the sheet on which a problem is printed to complete only that problem. Do not complete a problem on any other sheet.
7. The exam is open book, open electronics and open notes. Communication with other humans, except the instructor, regarding the exam, is not allowed. You may not use chat tools, or post to websites about the exam.
8. For each problem, **show or explain your work.** At most, partial credit will be given for a correct answer with no work shown. No credit will be given for an incorrect answer with no work shown.
9. Each page is worth an equal number of points.

1. (a) Give a formula for the speedup using Amdahl's law in terms of n and p.

   (b) If we calculate a speedup of 2 with 4 processors, and use of each processor costs $1 per hour, how much extra money are we paying to save a hour of computation time by using four processors instead of one?

   (c) Suppose the sequential part of a program accounts for 4% of the program's computation time. You are executing on a machine with n processors. How many processors would it take to double the speedup? Of course, your answer should be in terms of n.

2. (a) Consider the Test-and-Set spinlock on a system with several processors. Is it possible for the spinlock to spin on a cached memory location?

(b) Does decreasing the amount of time it takes, on average, for a processor to complete a critical section usually increase or decrease lock contention? Explain why.

(c) Let's say you have a linked list with a Test-And-Set (TAS) spinlock serializing access to it. Can you use this construction to implement a first-come-first-served (FCFS) lock for n threads? Explain how to do so, or why it is impossible.

3. (a) Consider the following tables. In a fully associative cache, which memory locations can load into each cache location? You may assume that each location in cache holds one memory location at a time.

Cache

Memory Location
0
4
8
12
16
20
24
28

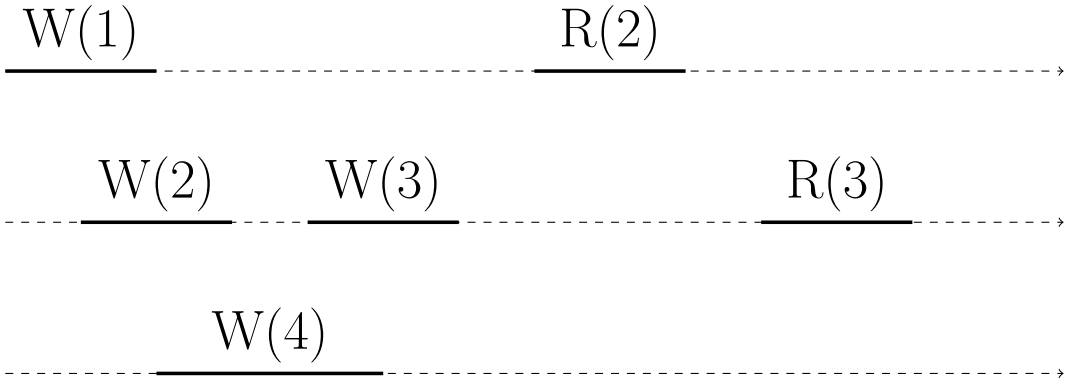
(b) Consider the following tables. In a 2-way associative cache, which memory locations can load into each cache location? You may assume that each location in cache holds one memory location at a time.

Cache

Memory Location
0
4
8
12
16
20
24
28

(c) Suppose we have a 32KB (1KB = 1024 bytes) cache with 256 cache entries. What is the cache line size?

4. Consider the following operations on a register.

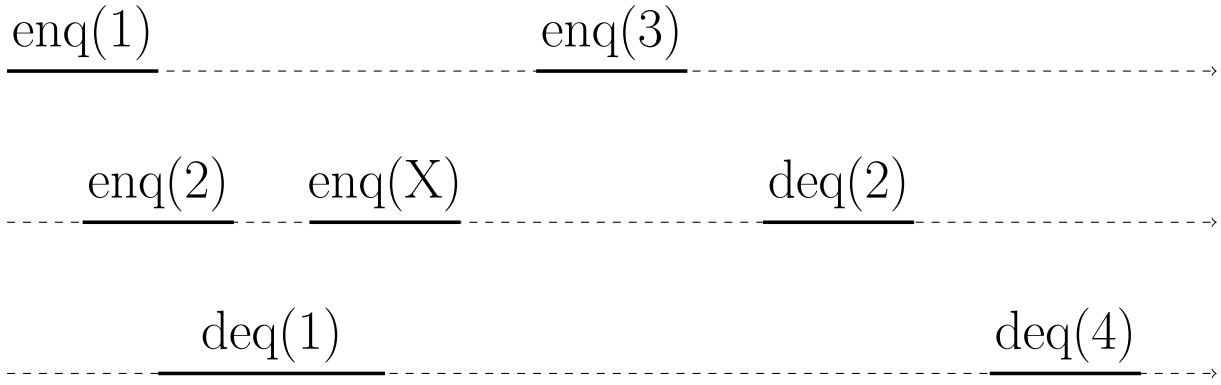


(a) Is the register safe? Provide a justification specific to this timeline of operations.

(b) Is the register regular? Provide a justification specific to this timeline of operations.

(c) Is the register atomic? Provide a justification specific to this timeline of operations.

5. Consider the following queue execution.

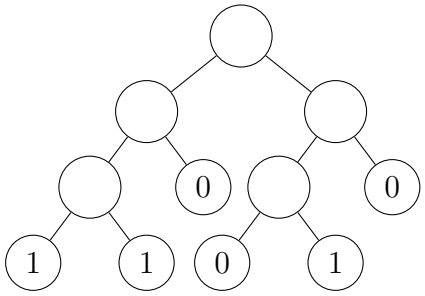


(a) What possible value(s), if any, could X be such that this execution would linearizable?

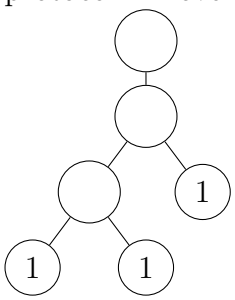
(b) Given what you determined above, in what order does a linearizable execution occur? You may simply number the diagram.

(c) Label all periods of quiescence on the diagram. How many are there? You may omit the period before and after the execution.

6. In the following consensus protocol tree, there are eleven states. Indicate, for each state, if that state is critical (C), bivalent (B), univalent(U) 1-valent (1-V), 0-valent (0-V), and final (F). Of course, some states may have more than one entry. Final states should be marked with an F and no other letter.



7. Consider the following tree. Is the tree a valid tree for a two-thread binary consensus protocol? Prove why it is or is not. You may use lemmas or theorems from the text.



8. Consider the following queue operations. Note carefully that this code differs from a similar example in the book and slides. If two threads are executing in parallel, and the first thread executes the enqueue one or more times, and the second thread executes the dequeue one or more times, is the implementation correct? Why or why not?

   You may assume that each line of code executes atomically, and that each line executes in the order in which it is listed in the code.

```
1   public void enq(Item x) {
2     if (tail-head == capacity)
3         throw new Exception();
4     tail++;
5     items[tail % capacity] = x;
6   }
7   public Item deq() {
8     if (tail == head)
9         throw new Exception();
10    head++;
11    Item item = items[head % capacity];
12    return item;
13  }
```

9. Consider the Exclusive Read (ER) PRAM Broadcast Algorithm.

   Suppose there are 16 processors (P_1 through P_{16}) and P_{16} has a value v = 3 that needs to be broadcast.

   (a) Show the steps (and during step 2, the iterations) required to store the value in P_{16} in all processors.

   (b) Let's say that during iteration 2 (where i=2), P_2 suffers a failure such that it writes 4 instead of 3. Show how this change affects the final state of the memory by drawing the memory locations.

10. Consider a lock that may provide some of the properties in section 2.2 of the textbook. In each case, briefly explain your answer.

    (a) Is it possible for a lock to be deadlock free and **not** provide mutual exclusion?

    (b) Is it possible for a lock to be starvation free and **not** provide mutual exclusion?

    (c) Is it possible for a lock to be deadlock free and **not** be starvation free?

    (d) Is it possible for a lock to be starvation free and **not** be deadlock free?