CS 351 Exam 4, Fall 2010

Your name: _____

Rules

- You may use one handwritten 8.5 x 11" cheat sheet (front and back). This sheet and the attached diagram are the only resources you may consult during this exam.
- Include explanations and comments in your answers in order to maximize your partial credit. However, you will be penalized for giving extraneous incorrect information.
- You may use the backs of these pages if you need more space, but make it clear where to find your answer to each question.
- Unless otherwise specified, you do not need to work out the arithmetic on math problems. Just do enough algebra to set up an answer of the form: Answer = [arithmetic expression] [units]

Grade (instructor use only)

	Your Score	**Max Score**
Problem 1		10
Problem 2		20
Problem 3		15
Problem 4		25
Problem 5		30
Total		100

Problem 1: 10 points (2.5 each).

The mythical town of Rosa Santa is located on a major geological fault. Exactly once every 100 years, an earthquake strikes Rosa Santa and devastates the town's road system. It always takes a year of construction to restore the roads.

---

a) What is the MTTF of Rosa Santa's road system? (Be careful!)

b) What is the MTTR of Rosa Santa's road system?

c) What is the MTBF of Rosa Santa's road system?

d) What is the availability of Rosa Santa's road system?

Problem 2: 20 points.

This question concerns a magnetic disk with a per-request overhead of 0.1 ms, a seek time of 5 ms, a rotational speed of 7200 rpm, and a transfer rate of 90 MB/s.

---

a) [5 points] List the steps in processing a 4 MB read request to this disk.

b) [15 points] What will the average time to process a 4 MB read request be? Show how each part of your answer correlates to steps from part (a).

Problem 3: 15 points.

a) [3 points] Which has the lowest cost per GB: DRAM, flash, or magnetic disk?

b) [3 points] Which has the highest cost per GB: DRAM, flash, or magnetic disk?

c) [3 points] Julian Assange is buying webservers to host a browsable mirror of a 100 GB Wikileaks archive. The archive will not change; he just wants to provide read-only access to it as quickly and to as many users as possible. What storage technology should he use?

Your explanation should clearly show the advantage of your chosen technology vs. alternatives. For example, saying “disk is nonvolatile” is not helpful, since flash is too.

Regardless of your political beliefs, you will not receive credit for answers attempting to sabotage Mr. Assange’s operation with bad IT advice.

d) [3 points] Is the following code an example of polling, interrupts, or DMA?

```
while (status_bit != READY) {
     status_bit = some_register & STATUS_BIT_MASK;
}
```

e) [3 points] The first step in running general-purpose code on a graphics processor is copying your data from system RAM to the GPU's memory. What I/O technique do you think is used for this operation? Justify your answer.

Problem 4: 25 points.

For this problem, consider a dataset large enough to fill eight (8) 1-terabyte disks.

A single small request on one of these disks takes 10 ms. A single large request takes 30 ms (10 ms of latency and 20 ms of transfer time).

---

a) [5 points] You are configuring a RAID 5 array with odd parity. If the original data bytes in a given position are as follows, what is the parity byte?

```
Disk 0: 1 0 0 0 0 1 1 0
Disk 1: 0 1 1 1 0 0 0 1
Disk 2: 0 1 1 1 0 0 0 1
Disk 3: 1 0 1 1 0 1 1 0
Disk 4: 0 0 1 1 1 0 0 0
Disk 5: 1 1 1 0 0 1 1 0
Disk 6: 1 1 1 0 1 1 1 0
Disk 7: 0 0 1 1 0 1 0 1

Parity byte:
```

b) [10 points] How many additional disks would you need to buy to configure a RAID-1 array?

On average, how long would it take to do 1000 small reads on this RAID-1 array?

On average, how long would it take to do 1000 small writes on this RAID-1 array?

On average, how long would it take to do a single large read on this RAID-1 array?

On average, how long would it take to do a single large write on this RAID-1 array?

c) [10 points] How many additional disks would you need to buy to configure a RAID-5 array?

On average, how long would it take to do 1000 small reads on this RAID-5 array?

On average, how long would it take to do 1000 small writes on this RAID-5 array?

On average, how long would it take to do a single large read on this RAID-5 array?

On average, how long would it take to do a single large write on this RAID-5 array?

Problem 5: 30 points.

---

a) [5 points] You are running a webserver that uses individual threads to handle individual users' requests as they come in. Would you rather have the latest multicore CPU or the latest GPU for this server? Explain.

b) [5 points] You are a physicist simulating a 1D array of particles at discrete timesteps using CUDA on a GPU. Your code looks something like:

```
___global___ void SimulateElectricField(float* E, float* E_old, float* H) {

        int myId = threadIdx.x;

        E[myId] = E_old[myId] + (H[myId] – H[myId-1]);
}
```

Why is there no loop in this code?

Remember that the GPU memory hierarchy must be managed explicitly; there is no automatic caching. Where is there an opportunity to exploit spatial or temporal locality in this code? Be specific about how you would optimize this code for the GPU memory hierarchy. (No need to write code, just explain what you would do.)

c) [5 points] What's the problem with this code?

```
...
int max = A[0];
#pragma omp parallel for
for (int i=1; i < N; i++)
     max[i] = max[i] > A[i] ? max : A[i];

cout << max << endl;
```

d) [5 points] What's the problem with this code?

```
int max = A[0];
#pragma omp parallel for private(max)
for (int i=1; i < N; i++)
     max[i] = max[i] > A[i] ? max : A[i];

cout << max << endl;
```

e) [10 points] Draw the dependency graph for the following snippet of code. Assume that all inputs are passed by value. Label each statement using the value on the left-hand side of the assignment (so the first statement is A, the second is D, etc.)

```
A = f(B, C);
D = g(B, C)
E = f(A, D);
C = g(B);
F = f(A, C, D);
```

Are there any constraints on the order of these operations that are not shown in the dependency graph?

If we have infinite parallel resources and the cost of communication is zero, how long will this code take to execute?