CPSC 320 Sample Midterm 2
November 2010

Name: ____________________ Student ID: ____________________
Signature: ____________________

- You have 50 minutes to write the 5 questions on this examination. A total of 40 marks are available.
- **Justify all of your answers.**
- You are allowed to bring in one hand-written, double-sided 8.5 x 11in sheet of notes, and nothing else.
- Keep your answers short. If you run out of space for a question, you have written too much.
- The number in square brackets to the left of the question number indicates the number of marks allocated for that question. Use these to help you determine how much time you should spend on each question.
- Use the back of the pages for your rough work.
- **Good luck!**

Question	Marks
1	
2	
3	
4	
5	
Total	

UNIVERSITY REGULATIONS:

- Each candidate should be prepared to produce, upon request, his/her library card.
- No candidate shall be permitted to enter the examination room after the expiration of one half hour, or to leave during the first half hour of the examination.
- CAUTION: candidates guilty of any of the following, or similar, dishonest practices shall be immediately dismissed from the examination and shall be liable to disciplinary action.
  1. Having at the place of writing, or making use of, any books, papers or memoranda, electronic equipment, or other memory aid or communication devices, other than those authorised by the examiners.
  2. Speaking or communicating with other candidates.
  3. Purposely exposing written papers to the view of other candidates. The plea of accident or forgetfulness shall not be received.
- Candidates must not destroy or mutilate any examination material; must hand in all examination papers; and must not take any examination material from the examination room without permission of the invigilator.

[6] 1. Short Answers

[3] a. If we can not use a recursion tree to prove a tight bound on the value of a function $T(n)$ defined by a recurrence relation, can we use the Master theorem instead? Explain why or why not.

[3] b. The first (and only) result we looked at in our outline of the proof of the Master theorem stated that

$$T(n) = n^{\log_b a} + \sum_{i=0}^{t-1} a^i f(n/b^i)$$

Explain where the term $a^i f(n/b^i)$ came from, by relating each factor to the tool we used to prove the lemma.

[5] 2. In class, we looked at an implementation of a binary counter, and proved that if we use as potential function $\Phi(D_i)$ the number of 1 bits in the counter, then the amortized cost of an INCREMENT operation is in $\Theta(1)$. Suppose instead that we had used the number of 0 bits in the counter as the potential function (note: this breaks the requirement that $\Phi(D_0) = 0$, so it would not be a valid choice). What would the amortized cost of INCREMENT be in this case?

[14] 3. Given a set T of n teams competing in some sport, a *round-robin tournament* is a collection of games in which each team plays each other team exactly once. We can describe such a schedule by an array of triplets, where the triplet (d, i, j) means that team i will play against team j on day d. A schedule is *reasonable* if no team plays more than one game per day, and *optimal* if it uses the smallest number of days out of all reasonable schedules.

[8] a. Design a **divide-and-conquer** algorithm that takes as input an integer n, and produces an optimal schedule as long as n is a power of 2.

[3] b. Prove that your algorithm produces an optimal schedule whenever n is a power of 2.

[3] c. Analyze the running time of the algorithm you described in your answer to part (a).

[6] 4. Write a recurrence relation that describes the worst-case running time of the following algorithm as a function of n. Recall that the call `BinarySearch(A, first, last, x)` runs in $O(\log(\texttt{last} - \texttt{first} + 1))$ time.

```
Algorithm Armadillo(A, first, n, x)
   y ← 1
   s ← ⌊√n⌋
   if (s > 1) then
      while (first + s < n) do
         y ← y * Armadillo(A, first, s, x) +
                BinarySearch(A, first, first + s - 1, x)
         first ← first + s
      endwhile
   endif
   return y
```

[9] 5. Prove upper and lower bounds on the function $T(n)$ defined by

$$T(n) = \begin{cases} 2T(3n/4) + 2T(n/4) + n^3 & \text{if } n \geq 4 \\ 1 & \text{if } n \leq 3 \end{cases}$$

Your grade will depend on the quality of the bounds you provide (that is, showing that $T(n) \in \Omega(1)$ and $T(n) \in O(100^n)$, while true, will not give you many marks).