CPSC 320 Sample Midterm 2
November 2011

Name: ____________________ Student ID: ____________________
Signature: ____________________

- You have 60 minutes to write the 5 questions on this examination.
  A total of 45 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 UBC 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.

[10] 1. Disjoint Sets Structures

[3] a. What is the Disjoint Sets data structure used for in Kruskal's minimum spanning tree algorithm?

[3] b. Intuitively (that is, without mentioning amortized costs or potential functions) what is the purpose of performing path compression during every `find()` operation in a Disjoint Sets data structure?

[4] c. In our amortized analysis of the Disjoint Sets data structure, we defined a node as *close*, *far* or *very far* depending on the relationship between its rank and the rank of its parent. We then analyzed the amortized cost of a `find(N)` operation, and argued that if this operation looks at l nodes, then its amortized cost is in $O(\log^* n)$, where n is the total number of nodes in the data structure. Explain briefly how we obtained this $\log^* n$ term.

[9] 2. Recurrence relations

[4] a. Explain how one should prove a tight bound on the solution to the following recurrence relation:

$$T(n) = \begin{cases} 5T(\lceil n/4 \rceil + 8) + \Theta(n) & \text{if } n \geq 12 \\ \Theta(1) & \text{if } n \leq 11 \end{cases}$$

Do not prove this bound; simply state how one should prove it.

[5] b. Write a recurrence relation that describes the running time of the following algorithm on an input array with n elements:

```
Algorithm Recursive(A, first, n)
  if (n > 1) then
    Recursive(A, first, n/2)
    Recursive(A, first + n/2, n/3)
    Recursive(A, first + n - n/4, n/4)
    LinearTimeAlgorithm(A, first, n/2, first + n/2, n - n/2 - n/4)
    LinearTimeAlgorithm(A, first, n - n/4, first + n - n/4, n/4)
```

You may assume that `/` performs integer division (that is, `x/y` returns the floor of x/y), and that a call to `LinearTimeAlgorithm`(A, first, n1, second, n2) runs in $O(n1 + n2)$ time. Simplify the recurrence relation you write as much as possible. You do not need to write floors and ceilings.

[8] 3. In class, we described an algorithm that finds both the minimum and the maximum elements of an array using at most $3\lfloor n/2 \rfloor$ comparisons (you do not need to remember the details of the algorithm for this problem; recall however that given two pairs (x_1, y_1), (x_2, y_2) where $x_1 \leq y_1$ and $x_2 \leq y_2$, we can determine the minimum and maximum elements of the two pairs, that is $(\min\{x_1, x_2\}, \max\{y_1, y_2\})$, using 2 comparisons).

Describe a divide-and-conquer algorithm that takes as input an array `A`, and two integers `first` and `last`, and returns the pair (x, y) where x is the smallest element from `A[first]` to `A[last]`, and y is the largest element from `A[first]` to `A[last]`.

[9] 4. Consider the following algorithm:

```
Algorithm Mysterious(array)
  accumulator ← 0
  for i ← 0 to length[array] - 1 do

    while (accumulator > 0 and compute(accumulator, array[i]) > 0)
      accumulator ← accumulator - 1

    if (array[i] is even) then
      accumulator ← accumulator + floor(log(i+1))

  return accumulator
```

Use the potential method to prove that this algorithm runs in $O(n \log n)$ time where $n =$ `length[array]`. You may assume that the function `compute()` runs in $\Theta(1)$ time. Hints:

- Think of the state of the algorithm at the end of the i$^{\text{th}}$ iteration as D_i.
- Use the value of `accumulator` at the end of the i$^{\text{th}}$ iteration as the potential of D_i.

Do not forget to show that Φ is a valid potential function.

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

$$T(n) = \begin{cases} T(n/2) + 2T(n/3) + T(n/6) + n^2 & \text{if } n \geq 6 \\ 1 & \text{if } n \leq 5 \end{cases}$$

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).