CPSC 320 Sample Midterm 1
October 2014

Name: ______________________ Student ID: ______________________

Signature: ______________________

– You have 65 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.

[13] 1. Short Answers

[3] a. In the version of the Gale-Shapley stable matching algorithm where women propose, is it possible for every man to get his worst possible partner? If so, what preference lists would result in that (unfortunate) outcome?

[3] b. What does each node of a decision tree represent?

[3] c. In case 3 of the proof of the theorem that we later used to show that the greedy caching strategy is optimal, we assumed that S_{FF} evicts an element e from the cache, whereas S_j ejects a different element f. We then stated that one of three things **must** happen before a request for e comes along:

- A request for f, where S_j evicts e.
- A request for f, where S_j evicts an element other than e.
- A request for some other element x, where S_j evicts e.

Why must one of these situations occur before any request for e?

[4] d. Consider the following two functions from $\mathbf{N}$ into $\mathbf{R}^+$:

$$f(n) = \begin{cases} 4n^3 & \text{if } n \text{ is even} \\ n^2/5 & \text{if } n \text{ is odd} \end{cases} \qquad g(n) = \begin{cases} n^3/6 & \text{if } n \text{ is even} \\ 2n^2 & \text{if } n \text{ is odd} \end{cases}$$

What is the relationship between f and g in terms of asymptotic notation? That is, is $f \in o(g)$, $f \in O(g)$, $f \in \Theta(g)$, $f \in \Omega(g)$, $f \in \omega(g)$, several of these, or none of them? Justify your answer.

[5] 2. Consider a problem $\mathcal{P}$ where, for an input of size n, there are 4^n possible answers. Prove as good a lower bound as possible on the worst-case running time of an arbitrary comparison-based algorithm for $\mathcal{P}$.

[5] 3. Ms. Ejiul, an alien computer scientist, designed a greedy algorithm that solves a graph problem by adding vertices and edges to her solution one at a time. She then proved that given a (not necessarily optimal) solution H that adds the same first j vertices and edges as her greedy algorithm, she can modify it to obtain a solution H' that is at least as good as H, and adds the same first $j+1$ vertices and edges as her greedy algorithm.

Is Ms. Ejiul's greedy algorithm optimal? Why or why not?

[8] 4. Recurrence relations

[4] a. Write a recurrence relation that describes the worst-case running time of the following algorithm as a function of n. You may ignore floors and ceilings. Note: I do not believe that this algorithm computes anything useful, so don't waste any time trying to understand what it does.

```
Algorithm ComputeTwo(A, first, n)
prod ← 1
if n ≥ 25 then
  for i ← 1 to ⌊√n⌋ do
    n ← n - 5
    prod ← prod * ComputeTwo(A, first, n)
  endfor
endif
return prod
```

[4] b. Write an algorithm whose running time can be described by the recurrence:

$$T(n) = \begin{cases} T(n/5) + 2T(n/4) + \Theta(n) & \text{if } n \geq 5 \\ \Theta(1) & \text{if } n < 5 \end{cases}$$

Your algorithm does not need to compute anything meaningful.

[14] 5. You and a group of your friends want to play tug-of-war (two teams pulling on opposites sides of a rope). Being the only computer scientist in the group, you have been asked to design an algorithm to build two teams A and B that are as equal as possible. You formalize the problem as follows:

- Each person i has a strength s_i.
- You would like the sum of the strengths of the people on team A to be as close to equal as possible to the sum of the strengths of the people on team B. That is, you want to minimize

$$\left| \sum_{i \in A} s_i - \sum_{j \in B} s_j \right|$$

Note that the two teams do not need to have the same number of people.

[8] a. Describe a greedy algorithm to build the teams A and B. Your algorithm does not need to always succeed at minimizing the difference in total strength between the two teams, but it should make a good attempt at it.

[3] b. Analyze the time complexity of your algorithm from part (a). Specify only as much of your implementation (for instance, data structures) as needed for your analysis. There is no need to get the most efficient possible implementation of your algorithm, but excessively pessimistic analyses (e.g. $O(n^4)$ when an algorithm could be implemented to run in $O(n \log n)$ time) will not get full marks.

[3] c. Give an example where your algorithm will not return the optimal solution (the one that minimizes the strength difference between the two teams).