Midterm Exam
CS 361, October 2010

This is a 75-minute exam. No calculators, notes, books, internet, phones, pagers, or other devices may be used. Write your answers in the space provided. Be concise! There are 6 problems total, each worth 20 points. Your lowest score will be dropped. Put your name at the top right of this page!

Problem 1. Multiple Choice: +4 points for each right answer, 0 for wrong answers. Clearly indicate your choice. For this section only, you do not need to justify your work.

1. Which best defines "The algorithm runs in time $O(n^3)$"?

   (a) The running time is always Cn^3 for some constant C.

   (b) The running time is at least Cn^3 for some constant C.

   (c) The running time is at most Cn^3 for some constant C.

   (d) The running time is sometimes, but not always, Cn^3, for some constant C.

2. Which of the following contradicts the statement, "The worst-case running time of the algorithm is $\Omega(n^2)$"?

   (a) The algorithm runs for $O(1)$ steps on some inputs.

   (b) The algorithm runs for $O(n)$ steps on no inputs.

   (c) The worst case running time is $O(n \log n)$.

   (d) The worst case running time is $O(2^n)$.

   (e) The worst case running time is $\Omega(n^3)$.

3. In defining the Stable Matchings problem, how did we proceed?

   (a) We said a matching is stable if no man wants to propose to someone else's wife.

   (b) We said a matching is stable if and only if every man and woman gets paired with their first choice.

   (c) We first defined what an "instability" is, and said a stable matching is one with no instabilities.

   (d) We defined the output of the Gale-Shapley algorithm to be a stable matching.

4. In the Gale-Shapley algorithm, run with n men and n women, what is the maximum number of times any woman can be proposed to?

   (a) $\Theta(1)$

   (b) $\Theta(n)$

   (c) $\Theta(n \log n)$

   (d) $\Theta(n^2)$

   (e) $\Theta(2^n)$

5. What does it mean for a graph algorithm to run in linear time? Assume the graph has n vertices and m edges.

   (a) The worst case running time is $O(n + m)$.

   (b) The worst case running time is $O(n^2)$.

   (c) The worst case running time is $O(n)$.

   (d) The worst case running time is $O(m)$.

Problem 2. For each of the following, say whether f is $O(g)$, $\Omega(g)$, $\Theta(g)$, or none of the above. **Justify your answers.**

1. $f(n) = \frac{n(n-1)}{2}$, $g(n) = n^2 + 2n$.

2. $f(n) = n^{\log(n)}$, $g(n) = (\log n)^n$.

3. $f(n) = 2^{\sqrt{n}}$, $g(n) = n^{(\log n)^2}$.

4. State the formal mathematical definition of $f = O(g)$.

Problem 3.

1. What is the definition of an *order relation*?

2. Give an example from class of an order relation.

3. What is the definition of an *equivalence relation*?

4. Give an example from class of an equivalence relation.

Problem 4. Suppose we are given an instance of the stable matching problem for which there is a man m who is the first choice of all women. Prove or give a counterexample: In any stable matching, m must be paired with his first choice.

Problem 5. Consider the following piece of pseudocode:

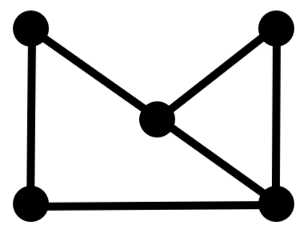
```
Given: S, a set of n numbers
total = sum of all elements of S
For all subsets T of S:
   val = sum of all elements of T.
   if (val == total - val) return TRUE
end For
return FALSE
```

1. Give the best upper bound you can on its running time. Justify your answer.

2. How can the code be improved to get a $\Theta(n)$ factor of speedup? Justify.

3. Suppose we only have time to do about a trillion ($10^{12} \approx 2^{40}$) operations. Roughly how large a value of n can we handle? Give answers for both the original code, and your improved version. (No, you don't need a calculator. I did say roughly.)

Problem 6. 1. Here is a drawing of a graph, G.



Find a spanning tree of G that is both a BFS tree and a DFS tree. Also, indicate where your tree has its root, as a BFS tree and as a DFS tree.

2. Suppose G is a **complete graph** on n nodes, that is, all $n(n-1)/2$ possible edges are present. Prove that, assuming $n \geq 3$, breadth-first search and depth-first search on G cannot produce the same spanning tree.