NAME: ____________________________

CMPSCI 611
Advanced Algorithms
Final Exam Fall 2012

A. McGregor

13 December 2012

DIRECTIONS:

- Do not turn over the page until you are told to do so.
- This is a *closed book exam.* No communicating with other students, or looking at notes, or using electronic devices. You may ask the professor to clarify the meaning of a question but do so in a way that causes minimal disruption.
- If you finish early, you may leave early but do so as quietly as possible. The exam script should be given to the professor.
- There are five questions. All carry the same number of points but some questions may be easier than others. Don't spend too long on a problem if you're stuck – you may find that there are other easier questions.
- The front and back of the pages can be used for solutions. There is also a blank page at the end that can be used. If you are using these pages, clearly indicate which question you're answering.
- The exam will finish at 12:30 pm.
- Good luck!

1	/10
2	/10
3	/10
4	/10
5	/10
Total	/50

Question 1 (True or False): For each of the following statements, indicate whether the statement is true or false by circling the appropriate option. It is not necessary to show working.

1. Given an unweighted graph, it is possible to find the length of the shortest path between every pair of vertices in polynomial time.

   - **TRUE**
   - FALSE

2. For any satisfiable instance ϕ of 3-SAT with more than one clause, a random assignment of the variables will satisfy ϕ with probability at least 7/8.

   - TRUE
   - **FALSE**

3. If $T(1) = 1$ and $T(n) = 2T(n/2) + n$ then $T(n) = \Theta(n \log n)$.

   - **TRUE**
   - FALSE

4. For any graph $G = (V, E)$, if $V' \subset V$ is a vertex cover then $V \setminus V'$ is an independent set.

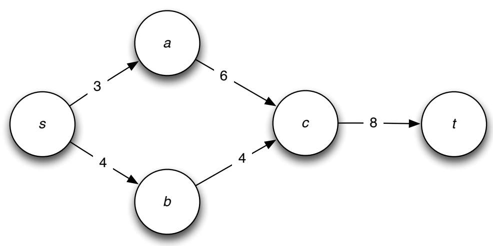
   - **TRUE**
   - FALSE

5. For any random variable X which is never negative, $\mathbb{P}\left[X < 10 \cdot \mathbb{E}\left[X\right]\right] \geq 9/10$.

   - **TRUE**
   - FALSE

Question 2 (Flows and Linear Programming):

1. What is the maximum s-t flow in the following network where labels on the edges indicate the capacities of the edges?



   ANSWER: The size of the maximum flow is 7.

2. State the Max-Flow/Min-Cut theorem.

   ANSWER: The size of the maximum flow equals the capacity of the minimum cut.

3. TRUE or FALSE: There exists a polynomial time algorithm for solving linear programs.

   - **TRUE**
   - FALSE

In the next two parts of the question, we consider the following linear program.

$$\begin{aligned} \text{maximize} \quad & 2x_1 + x_2 \\ \text{subject to} \quad & x_1 + x_2 \leq 3\ ,\ x_1 \leq 2\ ,\ x_2 \leq 2\ ,\ x_1, x_2 \geq 0 \end{aligned}$$

4. Draw the feasible region for the linear program. What is the optimal value?

   ANSWER: The optimal solution is 5 and occurs when $x_1 = 2$ *and* $x_2 = 1$.

5. Write the dual of the above linear program. What is the optimal value of the dual?

   ANSWER: The dual is minimize $3y_1 + 2y_2 + 2y_3$ *subject to* $y_1 + y_2 \geq 2, y_1 + y_3 \geq 1, y_1 \geq 0, y_2 \geq 0,$ *and* $y_3 \geq 0$. *The optimal value if also 5.*

Question 3 (Matchings): Consider a complete bipartite graph $G = (L \cup R, E)$ with nodes $L = \{l_1, \ldots, l_n\}$, $R = \{r_1, \ldots, r_n\}$, and edges $E = \{(l_i, r_j) : i, j \in \{1, 2, \ldots, n\}\}$. Suppose each edge $e \in E$ has some weight $w_e \geq 0$. The minimum weight matching problem is to find a matching M with n edges such that $\sum_{e \in M} w_e$ is minimized.

1. Prove that the minimum weight matching problem can be solved in polynomial time. You may use the fact that the maximum weight matching problem (i.e., find a matching that maximizes $\sum_{e \in M} w_e$) can be solved in polynomial time.

   ANSWER: Define $w^* = \max w_e$ and let $w'_e = w^* - w_e$. Then the weight of a matching M with respect to w'_e equals $\sum_{e \in M} w'_e = nw^* - \sum_{e \in M} w_e$. Hence, finding the maximum matching with respect to the weights w'_e yields the minimum weight matching with respect to the weights w_e. Hence the minimum weight matching can be solved in polynomial time.

2. Suppose each node v has a weight w_v such that $w_{l_1} < w_{l_2} < \ldots < w_{l_n}$ and $w_{r_1} < w_{r_2} < \ldots < w_{r_n}$. If the weight of an edge $e = (l, r)$ is defined as $w_e = |w_r - w_l|$, what is the weight of the minimum weight matching? Prove your answer is correct. **Hint**: What could you do if a matching contained edges (l_i, r_k) and (l_j, r_ℓ) for some $i < j$ and $k > \ell$?

   ANSWER: The weight of the minimum cost matching is $\sum_{i \in [n]} |w_{r_i} - w_{l_i}|$, i.e., each node l_i is paired with r_i. In any other matching there exists a pair of "crossing" edges (l_i, r_k) and (l_j, r_ℓ) for some $i < j$ and $k > \ell$. Replacing these edges by the edges (l_i, r_ℓ) and (l_j, r_k) reduces the weight and hence there can be no crossing edges. This follows because

   $$|w_{l_i} - w_{r_\ell}| + |w_{l_j} - w_{r_k}| \geq \max(w_{l_i}, w_{r_\ell}, w_{l_j}, w_{r_k}) - \min(w_{l_i}, w_{r_\ell}, w_{l_j}, w_{r_k})$$

   whereas

   $$|w_{l_i} - w_{r_\ell}| + |w_{l_j} - w_{r_k}| \leq \max(w_{l_i}, w_{r_\ell}, w_{l_j}, w_{r_k}) - \min(w_{l_i}, w_{r_\ell}, w_{l_j}, w_{r_k}) \ .$$

Question 4 (Verifying Matrix Multiplication): In this question we will design a randomized algorithm for checking matrix multiplication. You may use the following result without proof: If $Q(x_1, \ldots, x_n)$ is a non-zero polynomial with degree at most d and if $r_1, \ldots, r_n$ are chosen randomly from $\{0, 1, 2, \ldots, s-1\}$, then $\mathbb{P}[Q(r_1, \ldots, r_n) = 0] \leq d/s$.

1. Give an example of Q with $n = 1, d = 2, s = 3$ such that the above inequality is tight.

   ANSWER: $Q(x) = x(x-1)$.

2. Your friend claims that when you multiply two $n \times n$ matrices A and B, the resulting matrix is C. To check her answer, you randomly choose $r_1, \ldots, r_n$ such that each r_i is equally likely to be 0 or 1 and all r_i are independent. This defines vector $r = (r_1, r_2, \ldots, r_n)^T$.

   (a) If $AB \neq C$, prove that $\mathbb{P}[ABr \neq Cr] \geq 1/2$.

   ANSWER: Let $D = AB - C$. *If* $AB \neq C$ *then* D *has some non-zero row say* $(d_1, d_2, \ldots, d_n)$. *But then* $Q(x_1, \ldots, x_n) = \sum_{i=1}^{n} d_i x_i$ *is a non-zero polynomial of degree* 1. *Hence,* $\mathbb{P}[Q(r_1, \ldots, r_n) \neq 0] \geq 1/2$. *If* $Q(r_1, \ldots, r_n) \neq 0$ *then* Dr *has a non-zero entry and hence* $ABr \neq Cr$.

   (b) Give upper and lower bounds on the smallest time to compute ABr?

   ANSWER: We need to read every entry of A *and* B *so we need at least* $\Omega(n^2)$ *time. We can find* $v = Br$ *in* $O(n^2)$ *arithmetic operations and then find* Av *in an additional* $O(n^2)$ *arithmetic operations.*

3. Design and prove the correctness of an efficient randomized algorithm that, given input A, B, C and a positive parameter $\delta < 1/2$, will output "same" if $AB = C$ and will output "different" with probability at least $1 - \delta$ if $AB \neq C$. The more efficient your algorithm, the more marks you receive. You may assume the results in Part 2.

   ANSWER: Consider running the algorithm in Part 2 $\log \delta^{-1}$ *times with independent values of* r. *If the algorithm ever observes an* r *with* $ABr \neq Cr$ *then output "different". If not return "same". Note that if* $AB = C$ *then* ABr *always equals* Cr. *But if* $AB \neq C$, *then we know* $ABr \neq Cr$ *with probability at least* $1/2$. *Hence the probability that* $ABr \neq Cr$ *for one of the* $\log \delta^{-1}$ *choices of* r *is at least* $1 - (1/2)^{\log \delta^{-1}} = 1 - \delta$.

Question 5 (Coloring Graphs): A t-coloring of a graph $G = (V, E)$ is a labeling $f : V \to \{1, 2, \ldots, t\}$ of the vertices such that if $(u, v) \in E$ then $f(u) \neq f(v)$, i.e., adjacent nodes receive different labels. We say a graph G is t-colorable if there exists a t-coloring for G.

1. Design a polynomial time algorithm that determines whether a graph is 2-colorable. Prove the approximation ratio. **Hint:** Consider performing a breadth first search.

   ANSWER: It suffices to check whether each connected component of the graph is 2-colorable since a graph is 2-colorable iff each connected component is 2-colorable. Let v be an arbitrary node in a connected component and let L_i be the set of nodes in this connected component that are a distance i from v. The sets L_i can be found using a BFS from v and define $L_0 = \{v\}$. Suppose there is a 2-coloring and let $f(v) = 1$. Then, all nodes in L_1 must have color 2 (because each is adjacent to v), all nodes in L_2 must have color 1 (because each is adjacent to a node in L_1 with color 2) etc. So nodes in L_i must have color 1 if i is even and color 2 if i is odd. Hence the connected component is 2-colorable iff there doesn't edge between nodes in L_i and L_j where i and j are both even or both odd.

2. Let D be a maximum degree of a graph. Design a polynomial time algorithm that $(D + 1)/3$ approximates the minimum value of t such that the graph is t-colorable. Prove the approximation ratio.

   ANSWER: If the graph is 1-colorable (i.e., there are no edges) or 2-colorable we can determine this find the exact value of t. Otherwise we may assume t is at least 3. Consider coloring nodes in an arbitrary order where the color chosen for node v is the minimum value in $\{1, 2, \ldots, D + 1\}$ that is not already used to color a neighbor. Since there are at most D neighbors of each node, there is always a color that can be used. Hence we color the graph using $D + 1$ colors when we know the optimum values is at least 3. This gives the required approximation ratio.