Final Exam

Instructions: read these first!

Do not open the exam, turn it over, or look inside until you are told to begin.

Switch off cell phones and other potentially noisy devices.

Write your full name on the line at the top of this page. Do not separate pages.

You may have to use the inequalities: $1 - x \leq e^{-x}$ and $(n/k)^k \leq \binom{n}{k} \leq (ne/k)^k$.

Read questions carefully. Show all work you can in the space provided.

This is a closed book, closed notes exam. Do not refer to any printed material or computational device, except your cheat sheet. Your cheat sheet can be a single 8.5 × 11 sheet of paper printed on one or both sides.

Where limits are given, write no more than the amount specified. The rest will be ignored.

Avoid seeing anyone else’s work or allowing yours to be seen.

Do not communicate with anyone but an exam proctor.

If you have a question, raise your hand.

When time is up, stop writing.

You can see your graded final (and pick up your graded HW4) on Mon Mar 23, 2-3pm, in CSE 4110.

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>
1. [15 points] Answer the following questions.
   
   (a) [6 points] Let $a$ be an (unsorted) array of $n$ elements, where $n$ is a multiple of 4. We call an element $x$ an approximate maximum if it is greater than or equal to $3n/4$ elements of $a$. Write down a Monte Carlo algorithm that outputs an approximate maximum of $a$ with probability $\geq 1/4$ and runs in constant time.

   (b) [3 points] Prove the correctness of your algorithm in Part (a).
(c) [6 points] Now suppose you would like the algorithm to succeed with probability \( \geq 1 - \delta \) (as opposed to probability \( \frac{1}{2} \)). Use the algorithm in Part (a) to write down a Monte Carlo algorithm which outputs an approximate maximum of \( a \) with probability \( \geq 1 - \delta \) and runs in \( O(\log(1/\delta)) \) time. Prove the correctness of your algorithm.
2. [15 points] Ad-hoc networks are made up of cheap, low-powered wireless devices, which can communicate within a limited range. Consider an ad-hoc wireless network where any two devices can communicate if and only if the distance between them is at most $D$. To improve reliability, we require each device $x$ to have $k$ potential backup devices, all within distance $D$ of $x$; we call these $k$ devices the backup set of $x$. Also, we do not want any device to be in the backup set of more than $b$ devices, say; otherwise, a single failure might affect a large fraction of the network.

Suppose we are given the communication distance $D$, parameters $b$ and $k$, and an array $d[1...n, 1...n]$ of distances, where $d[i, j]$ is the distance between device $i$ and device $j$. The following questions ask you to design and analyze an algorithm that either computes a backup set of size $k$ for each of the $n$ devices, such that that no device appears in more than $b$ backup sets, or correctly reports that no good collection of backup sets exists.

(a) [7 points] First, reduce this problem to a max-flow problem. Describe a max-flow problem (a directed, capacitated graph $G$, a source and sink pair) such that a good collection of backup sets exist if and only if the max-flow in $G$ has a certain size. What is this size?
(b) [5 points] Now, prove the correctness of your reduction. Show that a good collection of backup sets exist if and only if the max-flow in $G$ has a certain size.

(c) [3 points] What algorithm would you use to solve this max-flow problem? What is the running time of this algorithm as a function of $n$, $k$ and $b$?
3. [20 points] For each of the following statements, say whether they are correct or incorrect. If they are correct, provide a brief justification or proof; if they are incorrect, provide a brief justification or counterexample.

(a) [5 points] Let $G$ be an arbitrary flow network with source $s$, sink $t$ and positive integer capacities $c(e)$ on every edge $e$. Let $(A, B)$ be a minimum cut in $G$ with respect to the capacities $c(e)$. Now suppose we add 1 to every capacity; then $(A, B)$ is still the minimum cut with respect to the new capacities.

(b) [5 points] Professor A. F. Lake gives you a directed flow network $G = (V, E)$ along with edge capacities $c(e)$, and a flow $f(e)$, and claims that $f$ is a max flow in $G$. Then, his claim can be verified in $O(|V| + |E|)$ time.
(c) [5 points] Suppose we are given a Las Vegas algorithm $A$ whose expected running time is $T(n)$ on inputs of size $n$. Then, we can use $A$ to build a Monte Carlo algorithm $A'$ with the following property. Given an input of size $n$, $A'(n)$ returns the correct answer with probability $\geq 1/2$ and has worst case running time $2T(n)$.

(d) [5 points] Let $G = (L, R, E)$ be a bipartite graph. The following Algorithm $A$ for computing a matching $M$ in $G$ computes a maximum cardinality matching of $G$.

While there is an unmatched node $v$ in $L$ 
    If any neighbor $u$ of $v$ is unmatched, add $(v, u)$ to $M$