1. Let $A$ be a randomised algorithm with running time $O(T(n))$ and $F$ a function such that $A$ either returns $F(x)$ or \texttt{timeout} on any input $x$. Assume that there exists a constant $p > 0$ such that for all inputs $x$, $A$ returns $F(x)$ with probability at least $p$. Give an algorithm $A'$ that almost surely (i.e., with probability 1) returns $F(x)$ on input $x$ and whose expected running time is $O(T(n)/p) = O(T(n))$.

2. Let $a_1, a_2, \ldots, a_n$ be a list of $n$ distinct numbers. We say that $a_i$ and $a_j$ are inverted if $i < j$ but $a_i > a_j$. The \texttt{Bubblesort} algorithm works by swapping adjacent inverted numbers until there are no inverted numbers. Suppose that the input to \texttt{Bubblesort} is a permutation chosen uniformly at random from any of the $n!$ permutations of the $n$ distinct numbers. Determine the expected number of swaps performed by \texttt{Bubblesort}.

3. Consider the following algorithm \texttt{RandomSelect} for finding the $k$th smallest element of an unsorted set $S$ of size $n$:

\begin{verbatim}
RandomSelect(S, k)
   Pick an element $p \in S$ at random
   By comparing $p$ to each element of $S$, compute
   $S_1 := \{x \in S \mid x < p\}$
   $S_2 := \{x \in S \mid x > p\}$
   If $|S_1| = k - 1$ then output $p$
   If $|S_1| > k - 1$ then output \texttt{RandomSelect}(S_1, k)
   If $|S_1| < k - 1$ then output \texttt{RandomSelect}(S_2, k - |S_1| - 1)
\end{verbatim}

Let $T(n, k)$ denote the expected time (number of comparisons) required by \texttt{RandomSelect} to find the $k$th smallest element of a set of size $n$, and let $T(n) = \max_k T(n, k)$. Show that $T(n)$ is at most $4n$.

[\textit{Hint}: Establish a recurrence for $T(n)$.]}

4. The analysis of the algorithm for MAX-3-SAT showed that a random truth assignment satisfied a $7/8$-fraction of the clauses in expectation. Using Markov’s inequality, show that for $0 < \epsilon \leq 7/8$, repeating the randomized algorithm $t = O(1/\epsilon)$ times and taking the best of the $t$ solutions satisfies at least $(7/8 - \epsilon)$-fraction of the clauses with probability at least $1/2$.

5. A matching in a graph is a set of edges without common vertices. In the Maximum Bipartite Matching problem, we are given a bipartite graph $G(L \cup R, E)$ and we want to find a matching of
maximum cardinality. Consider the following randomised algorithm for this problem: Each edge is selected independently with probability $p$. All edges that have common points are discarded. Assume that the bipartite graph has $|L| = |R| = n$ and that every vertex has degree 3.

- What is the expected cardinality of the matching returned by the algorithm as a function of $p$?
- Find the value of $p$ that maximises the expected cardinality of the matching. What is the approximation ratio of this algorithm?
- Show how to derandomise the algorithm.

6. Consider the undirected graph $G(V,E)$ of a social network with $n$ members, where the nodes are the members of the network and edges indicate friendship. For a node $u$, let $d_u$ indicate the degree of $u$ (the number of its friends).

(a) Show that for a random variable $X$: $E[X^2] \geq E[X]^2$.

(b) Explain the “friendship paradox”, which roughly says that on expectation people have fewer friends than their friends have. More precisely, select a node $u$ uniformly at random and consider the expected number $x$ of the friends of its friends (defined formally as the sum of the degrees of $u$’s friends). If the friends of $u$ had exactly the same number of friends as $u$, this number $x$ would be $d_u^2$. Show that $x \geq E[d_u]^2$. The friendship paradox is that this is strict inequality for typical networks. Give a small network in which this is strict inequality.

(c) We want to pay a subset $P$ of the members of the network to post a message with the goal of influencing as many members of the network as possible. Let $S_P$ denote the set of influenced members:

$$S_P = \{v : \exists u \in P \text{ such that } [u,v] \in E\}.$$  

Note that it is possible for a node to get a payment and not to be influenced.

Finding the optimal subset $P$, among the subsets of a given cardinality, seems like a hard problem so we will run a randomised algorithm that selects nodes with probability proportional to their degree. Specifically, fix some parameter $w \in (0, 1/d)$, where $d$ is the maximum degree of the network. Consider a random subset $P$ of the network created by selecting each member $u$ with probability $wd_u$, independently.

Suppose that all the nodes of the network have the same degree $d$ and $w = \varepsilon/d^2$, for some constant $\varepsilon \in (0, d)$. Show that the algorithm achieves a constant approximation ratio, with respect to the optimal algorithm which selects the best $wdn$ nodes to maximise the number of influenced members.

(d) Explain how to use the method of conditional expectations method to derandomise the algorithm and obtain a polynomial-time deterministic algorithm that has the same or better performance than the randomised algorithm.