1. (a) The probability of a given time slot yielding a conflict is $\frac{1}{N}$, except for the last slot, which cannot give a conflict. So the probability of all time slots being conflict-free is $(1-\frac{1}{N})^{N-1} \approx e^{-1}$. **[8]**

   (b) Now, a given slot will cause a conflict if the slot following on Tuesday has one of 2 speakers. So the probability of one slot being conflict-free is $(1-\frac{2}{N})$, and for all slots it is $(1-\frac{2}{N})^{N-1} \approx e^{-2}$. **[8]**

   (c) This is the expected number of fixed points in a random permutation: P(fixed point at i) $= \frac{1}{N}$, so E[fixed points] $= N\frac{1}{N} = 1$. **[4]**

2. We have $\mu = \frac{N}{6}$, $\sigma^2 = Np(1-p) = \frac{5N}{36}$.

   (a) Markov is $P(X \geq \frac{N}{4}) \leq \frac{\mu}{\frac{N}{4}} = \frac{2}{3}$. **[6]**

   (b) Chebyshev: $P(X \geq \frac{N}{4}) \leq P(|X - \frac{N}{6}| \geq \frac{N}{4} - \frac{N}{6} = \frac{N}{12} = t\sigma) \leq \frac{1}{t^2}$. So solving $t\sigma = \frac{N}{12}$ we get $\frac{1}{t^2} = (\frac{12}{N})^2\sigma^2 = \frac{20}{N}$. We cannot halve this value to improve the bound, as the binomial distribution with $p = \frac{1}{6}$ is not symmetrical. **[7]**

   (c) Chernoff: $P(X \geq \frac{N}{4}) = P(X \geq (1+\delta)\mu) \leq (\frac{e^\delta}{(1+\delta)^{(1+\delta)}})^\mu \leq e^{-\delta^2\mu/4}$. Solving $(1+\delta)\mu = \frac{N}{4}$ gives $\delta = 0.5$, and the first bound is $e^{-0.018N}$ while the second bound is $e^{-N/96}$. **[7]**

3. (a) If we treat G as a multigraph, then each edge added is counted once, and we get $\binom{n}{2}$ expected edges.

   If adding a duplicate edge leaves just a single edge in the graph, then the probability of a single edge not being used is $p = (1 - \frac{1}{/}r)^r$, where $r = \binom{N}{2}$. So the expected number of edges being used is $k(1-p) \approx \binom{N}{2}(1-\frac{1}{e})$. (This second argument might not be needed here, depending on the interpretation of the question, but if not, we do need the same logic in part b) **[4]**

   (b) A perfect matching is a set of $N/2$ (disjoint) pairs of vertices, each of which will be contracted. Contraction will give a new graph with at most $\binom{N/2}{2}$ edges. Note that contracting an pair ab will also eliminate edges which become duplicate links to the new combined (ab) vertex, so more than simply $N/2$ edges get eliminated.

   After contraction, consider an edge between a pair of contracted vertices. If pairs ab and cd were contracted, then an edge in the contracted graph could have originally been ac,ad,bc, or bd. So the probability of the edge in the contracted graph not being used is $p = (1-\frac{4}{k})^k \approx e^{-4}$, and so the expected number of edges in G" is $\binom{N/2}{2}(1-e^{-4})$. **[8]**

   (c) After k contractions, the contracted graph must have C = $\binom{N/2^k}{2}$ edges to be a complete graph (that is, to have all possible edges). We must collect at least one of each of these edges in the random sampling of the original graph, with $r = \binom{N}{2}$ samplings. So we need $r > C\ln C + O(C)$, which yields $k = O(\ln\ln N)$. **[8]**

4. (a) $\phi(n) = n(1-\frac{1}{2})(1-\frac{1}{3}) = 2 \cdot 3^{k-1}$. **[4]**

(b) The order of a subgroup must divide $2 \cdot 3^{k-1}$. This means that orders $3^i 2^j$ are possible, where $0 \leq i \leq k-1$ and $j \in \{0,1\}$. That includes order $2 \cdot 3^{k-1}$, as Z_n^* is cyclic. **[8]**

(c) Let g be a generator for Z_n^*. All elements of Z_n^* are powers of g. Now an element, writing it as g^i, is a generator if and only if i is a generator in the additive group $Z_{\phi(n)}$. Otherwise g^i generates a subgroup. That means i is relatively prime to $\phi(n) = 2.3^{k-1}$, that is, i is neither a multiple of 2 nor 3. One third of the elements between 0 and $\phi(n)-1$ are not multiples of 2 or 3. Or to put it another way, the fraction of generators of the additive group $Z_{\phi(n)}$ is $\frac{\phi(\phi(n))}{\phi(n)} = \frac{2 \cdot 3^{k-2}}{2 \cdot 3^{k-1}} = \frac{1}{3}$.

(Using the approximation $\phi(N)/N > 1/\log N$ does not give an accurate result.) **[8]**

5. In this case, $h(M) = M^k \bmod n = (M \bmod n)^k \bmod n$. Since M is not restricted to be smaller than n, this cannot be a collision free (weakly or strongly) hash function since for any message M, the values $M + kn$ will be mapped to the same hash value for any k. On the other hand, it is a one-way hash function, since if you could find an x to match a given output y, you could also decrypt RSA...

6. (a) Y can send to the bank the string of bits corresponding to

$$Signed_Y(Signed_X(Bank, Amount, Y))$$

Including Y's name along with the bank's name and the amount means that X intended the check for Y. X's signature means that X intended to write the check. Y's signature means that Y intended to cash the check. To check that the person with the check is really Y, the bank could ask them to sign any random message M when they show up at the bank. After verifying that signature using Y's public key, the bank would know that the person is Y.

(b) X asks Y to generate a new and temporary RSA key, which creates a new and temporary identity which we will call $Z = (Z_{public}, Z_{private})$. Y then transmits (securely) the public part Z_{public} of this key to X. X creates a check which is $Signed_X(Bank, amount, Z_{public})$. X sends this check to Y, and Y signs the check to give $Signed_Z(Signed_X(Bank, amount, Z_{public}))$ showing that the person the check was written for accepted it (anyone can verify this using the public key in the check). Y goes to the bank with this check. The bank will be able to see that the check was intended for and accepted by whoever has Z's identity. Then the bank can issue a random challenge M. If Y signs M using the private key $Z_{private}$, the signature $Signed_Z(M)$ can be verified using the public key Z_{public} included in the check. This proves to the bank that this is the person the check was intended for.