Solutions for Sample Midterm 2

1. The number of primes $\le k$ is $\pi(k) \sim \frac{k}{\ln k}$, so the probability is $\pi(k)/k \sim 1/\ln k$.

2. **(a)** $1/3$

(b) $1/(3^2) = 1/9$.

3. **(a)** The probability of error is $\le 1 - 1/n$; we want to reduce it to $1/e$. If we do T trials then the probability of getting a "no" on every one when the answer is in fact "yes" is $\le (1-1/n)^T$. If we choose $T = n$, then we get $(1-1/n)^n \sim e^{-1}$.

(b) We need to reduce the probability of error from $1/e$ to $1/(e^{100})$, so we do 100 trials of the boosted algorithm in (a), or $100T$ trials of the original algorithm.

4. The probability that the first clause is not satisfied is $\Pr[x_1 = x_2 = 0] = 1/4$, so it is satisfied with probability $3/4$. Similarly, the second and third clauses are satisfied with probabilities $1/2$ and $3/4$, respectively. Thus the answer is $\mathrm{E}(X_1 + X_2 + X_3) = \mathrm{E}(X_1) + \mathrm{E}(X_2) + \mathrm{E}(X_3) = 3/4 + 1/2 + 3/4 = 2$, where X_i is the indicator random variable for satisfying the ith clause.

5. $\mathrm{E}(S_n) = n\mu$ and $\mathrm{Var}(S_n) = n\sigma^2$. Thus the quantity in question is $\frac{S_n - \mathrm{E}(S_n)}{\sqrt{\mathrm{Var}(S_n)}} = \frac{S_n - n\mu}{\sigma\sqrt{n}}$.

6. The proportion of heads $S_n = (X_1 + \ldots + X_n)/n$, where $X_i = 1$ if ith toss was heads and 0 otherwise. $\mathrm{E}(X_i) = 1/2$, and $\mathrm{Var}(X_i) = 1/4$, so $\mathrm{E}(S_n) = (n/2)/n = 1/2$ and $\mathrm{Var}(S_n) = \frac{n\mathrm{Var}(X_i)}{n^2} = \frac{1}{4n}$. Furthermore, by the Central Limit Theorem, S_n is approximately Normal. Thus, the answer is the standard deviation of S_n, which is $\sqrt{\mathrm{Var}(S_n)} = \frac{1}{2\sqrt{n}}$.

Since $\mathcal{H}$ is a 2-universal family, we have $\Pr[h(x) = h(y)] \le \frac{1}{|T|}$ for h chosen u.a.r. from $\mathcal{H}$. Since there are $|\mathcal{H}|$ hash functions in total, the number of those with $h(x) = h(y)$ must be $|\mathcal{H}| \cdot \Pr[h(x) = h(y)] \le \frac{|\mathcal{H}|}{|T|}$.

8. **(a)** $\mathrm{E}(|S|) = n/4$.

(b) For each edge e of G, $\Pr[e \text{ is inside } S] = \Pr[\text{both endpoints of } e \text{ are in } S] = 1/4^2$, and since there are $2n$ edges in G, we have $\mathrm{E}(X) = 2n \cdot 1/16 = n/8$.

(c) S' must be independent because otherwise there would be an edge e between a pair of vertices in S', which is impossible since one of e's endpoints would have been removed. Since at most one vertex is removed for each e inside S, we have $\mathrm{E}(\text{number of removed vertices}) \le \mathrm{E}(\text{number of edges inside } S) = n/8$, by part (b). So $\mathrm{E}(|S'|) = \mathrm{E}(|S|) - \mathrm{E}(\text{number of removed vertices}) \ge n/4 - n/8 = n/8$.

(d) Since $\mathrm{E}(|S'|) \ge n/8$, there must exist a set S' such that $|S'| \ge n/8$. Such a set S' must be independent by construction.

(e) The algorithm: generate S' as above and output it.

S' is always an independent set. Let's look at $\Pr[|S'| \ge n/16]$. Since there are n people, the random variable $Y = n - |S'|$ is non-negative. Furthermore, $\mathrm{E}(Y) = n - \mathrm{E}(|S'|) \le 7n/8$. Thus, $\Pr[|S'| < n/16] = \Pr[Y > n - n/16] \le \frac{\mathrm{E}(Y)}{n - n/16} \le \frac{7n/8}{15n/16} = 14/15$, where the second to last inequality is obtained by applying Markov's inequality to Y. Therefore, $\Pr[|S'| \ge n/16] = 1 - \Pr[|S'| < n/16] \ge 1 - \frac{14}{15} = 1/15$. So our algorithm does, in fact, output an independent set S' which has at least $n/16$ people with probability at least $1/15$.

9. **(a)** A vertex v is isolated if and only if none of the $n-1$ edges connecting it to the other vertices of G is present. The probability of this is $(1-p)^{n-1}$ since $1-p$ is the probability for the absence of a particular edge.

(b) $X = \sum_{i=1}^{n} X_i$ where $X_i = 1$ if the ith vertex is isolated, and $= 0$ otherwise. Thus $\mathrm{E}(X) = \sum_{i=1}^{n} \mathrm{E}(X_i) = n(1-p)^{n-1}$.

(c) $\ln \mathrm{E}(X) = \ln n + (n-1)\ln(1-p) \leq \ln n + (n-1)(-p) = (\ln n)(1 - \frac{n-1}{n}\frac{p}{(\ln n)/n})$. Since $p \gg \frac{\ln n}{n}$, we have $\frac{p}{(\ln n)/n} \to \infty$. Further, $\frac{n-1}{n} \to 1$, and so $(1 - \frac{n-1}{n}\frac{p}{(\ln n)/n}) \to -\infty$. Since also $\ln n \to \infty$, we have $\ln \mathrm{E}(X) \to -\infty$. Therefore, $\mathrm{E}(X) \to 0$.

(d) $\ln \mathrm{E}(X) = \ln n + (n-1)\ln(1-p) \geq \ln n + (n-1)(-2p) = (\ln n)(1 - 2\frac{n-1}{n}\frac{p}{(\ln n)/n})$. Since $p \ll \frac{\ln n}{n}$, we have $\frac{p}{(\ln n)/n} \to 0$. Further, $\frac{n-1}{n} \to 1$, and so $(1 - 2\frac{n-1}{n}\frac{p}{(\ln n)/n}) \to 1$. Since also $\ln n \to \infty$, we have $\ln \mathrm{E}(X) \to \infty$. Therefore, $\mathrm{E}(X) \to \infty$.

(e) If $p \gg \frac{\ln n}{n}$, we have by Markov's inequality $\Pr[G \text{ has isolated vertex}] \leq \mathrm{E}(X) \to 0$. Therefore $\Pr[G \text{ has isolated vertex}] \to 0$.

(f) If $p \ll \frac{\ln n}{n}$, we have $\Pr[G \text{ has no isolated vertices}] = \Pr[X = 0] \leq \Pr[|X - \mathrm{E}(X)| \geq |\mathrm{E}(X)|] \leq \frac{\mathrm{Var}(X)}{\mathrm{E}(X)^2} \to 0$, so $\Pr[G \text{ has no isolated vertices}] \to 0$ and $\Pr[G \text{ has isolated vertex}] \to 1 - 0 = 1$.

(g) We know that $\mathrm{Var}(X) = \mathrm{Var}(\sum_{i=1}^{n} X_i) = \sum_i \mathrm{Var}(X_i) + \sum_{i \neq j} \mathrm{Cov}(X_i, X_j)$, where X_i are the indicator variables for each vertex, and Cov() denotes covariance (as in lecture notes). We have $\mathrm{Var}(X_i) = (1-p)^{n-1}(1-(1-p)^{n-1})$, since $\Pr[i\text{th vertex is isolated}] = (1-p)^{n-1}$. Let us now compute $\mathrm{Cov}(X_i, X_j) = \mathrm{E}(X_i X_j) - \mathrm{E}(X_i)\mathrm{E}(X_j)$. We have, $\mathrm{E}(X_i X_j) = \Pr[X_i = X_j = 1] = \Pr[\text{both } i\text{th and } j\text{th vertices are isolated}]$. For the latter event to occur, it must be that the edge between i and j is missing, as are the $2(n-2)$ edges connecting i or j to the remaining $n-2$ vertices. Thus, $\Pr[X_i = X_j = 1] = (1-p)^{1+2(n-2)} = (1-p)^{2n-3}$, and $\mathrm{Cov}(X_i, X_j) = (1-p)^{2n-3} - ((1-p)^{n-1})^2 = (1-p)^{2n-3}(1-(1-p)) = p(1-p)^{2n-3}$.

We can now write $\mathrm{Var}(X) = n \cdot (1-p)^{n-1}(1-(1-p)^{n-1}) + n(n-1) \cdot p(1-p)^{2n-3}$, and $\frac{\mathrm{Var}(X)}{\mathrm{E}(X)^2} = \frac{n(1-p)^{n-1}(1-(1-p)^{n-1}) + n(n-1)p(1-p)^{2n-3}}{n^2(1-p)^{2n-2}} = \frac{1-(1-p)^{n-1}}{n(1-p)^{n-1}} + \frac{(n-1)p}{n(1-p)}$. We know from (d) that $\mathrm{E}(X) = n(1-p)^{n-1} \to \infty$ when $p \ll \frac{\ln n}{n}$. Therefore, the first term, $\frac{1-(1-p)^{n-1}}{n(1-p)^{n-1}} \to 0$, since the numerator is between 0 and 1. What about the second term $\frac{(n-1)p}{n(1-p)}$? We have $\frac{n-1}{n} \to 1$, and $\frac{p}{1-p} \to 0$ since $p \ll \frac{\ln n}{n}$ and $\frac{\ln n}{n} \to 0$. Therefore, $\frac{\mathrm{Var}(X)}{\mathrm{E}(X)^2} \to 0 + 1 \cdot 0 = 0$, as $n \to \infty$, if $p \ll \frac{\ln n}{n}$.

10. **(a)** The polynomials Q_X and Q_Y will be identical if and only if their representations as products $(z - \alpha_1)\ldots(z - \alpha_n)$ are the same up to a permutation, that is, if and only if $X = Y$. Thus, we simply use the Schwartz-Zippel algorithm to check whether $Q_X - Q_Y \equiv 0$. When $X = Y$, the polynomials will be identical and the output will always be "yes". If $X \neq Y$, the output will be "yes" with probability at most $d/|S|$, where $d = n$ is the degree of the polynomials and S is the set from which random values for z are drawn. Taking a set with $|S| \geq 2n$, say $S = \{1, 2, \ldots, 2n\}$, we will have a false "yes" with probability at most 1/2.

(b) The running time is $O(n)$, since that's how long it takes to evaluate $Q_X(z)$ and $Q_Y(z)$ for any value of z (n subtractions and $n-1$ multiplications).

(c) The above algorithm is just comparing two numbers, $Q_X(r)$ and $Q_Y(r)$, where $z = r$ is a (random) value for z. Each of these numbers has at most $b = n\log m$ bits, because $|Q_X(z)| \leq m^n$. So we can use the Alice and Bob trick to reduce this to comparing two much smaller fingerprints, of only $O(\log b) = O(\log n + \log\log m)$ bits. The fingerprint of a number is just the number mod p, where p is a prime chosen u.a.r. from $\{1, 2, \ldots, k\}$, where $k = O(b \log b)$; so p has only $O(\log b)$ bits. From our analysis in class, this gives only a small probability of error in the comparison (and hence a small *additional* probability of error in the above algorithm). To implement this scheme, we simply perform *all* the arithmetic mod p: this ensures that no *intermediate* integers appearing in the calculation require more than $O(\log n + \log\log m)$ bits, as required. (Note that the input integers x_i and y_i actually require $O(\log m)$ bits; the question is slightly misleading here.)

Note that it is *not* enough to simply fingerprint the factors $(z - x_i)$ and $(z - y_i)$. When they are multiplied together, larger numbers may appear.