Assignment 1
CS289: Pseudorandomness, Spring 2017
Due: May 3

Guidelines for submitting the solutions:

- It is strongly recommended to use LaTeX or other word processing software for submitting the homework. This is not mandatory but will be helpful both for you and for us. If submitting electronically, email the solutions to me before midnight the day they are due. You also **have to scan your document to a PDF** and cannot submit image files (they are very hard to read when you print them out).
- All problem numbers below refer to the text Pseudorandomness by Salil Vadhan.

1. For a simple graph $G = (V, E)$, the Tutte matrix in variables $x = (x_{ij} : i < j)$ is the skew symmetric matrix $T(x) \in \mathbb{R}^{n \times n}$ defined as follows:

$$T(x)_{ij} = \begin{cases} x_{ij} & i < j \\ -x_{ij} & i > j \\ 0 & i = j \end{cases}.$$

   Develop a randomized algorithm for checking if a given graph G has perfect matching by using the Tutte matrix. You have to prove the correctness of your algorithm. [3 points]

2. Problem 3.6. [3 points]

3. The following presents a technique that is useful in several contexts for developing deterministic algorithms from randomized ones. Suppose we have a series of events $A_1, \ldots, A_N \subseteq \{0,1\}^r$ and a distribution μ on $\{0,1\}^r$. If $\mu(A_1) + \ldots + \mu(A_N) \leq 1/2$, then a random sample $x \sim \mu$ will likely not be in $A \equiv \cup_i A_i$ (as in the *probabilistic method*). Our goal is to find such a x deterministically.

   Suppose there are functions $\varphi_i : \{0,1\}^i \to [0,1]$ for $i = 1, \ldots, r$ such that

   - Each function φ_i is efficiently computable and φ_0 is a constant with $\varphi_0 < 1$.
   - For each i and $a_1, \ldots, a_i \in \{0,1\}$,

$$\Pr_{x \leftarrow \mu}[x \text{ satisfies } A | x_1 = a_1, \ldots, x_i = a_i] \leq \varphi_i(a_1, \ldots, a_i).$$

   - For each i and $a_1, \ldots, a_i \in \{0,1\}$, there exists $a \in \{0,1\}$ such that $\varphi_{i+1}(a_1, \ldots, a_i, a) \leq \varphi_i(a_1, \ldots, a_i)$.

Solve the following:

(a) Applying the Chernoff bound and the union bound show the following. [1 point]
Let $S_1, \dots, S_m \subseteq [n]$ be sets of size k each, where $k \geq 100 \log m$. Then, there exists $S \subseteq [n]$ such that

$$\max_{1 \leq i \leq m} ||S_i \cap S| - k/2| \leq k/4.$$

(b) Use the method of pessimistic estimators to give a **deterministic** polynomial-time algorithm to find a S as above. [2 points].
(Hint: Let μ denote the uniform distribution over subsets of $[n]$. For $1 \leq j \leq m$, let A_j be the event that $||S_j \cap S| - k/2| \leq k/4$. Then, $\mu(\cup_j A_j) \leq \sum_j \mu(A_j)$. Go back to the proof of the Chernoff bound based on the exponential generating functions we saw in class (to bound each $\mu(A_j)$) and see if you can find suitable pessimistic estimators from the proof.)

4. (Lower bound for k-wise independent spaces) Let μ be a k-wise independent distribution on $\{1, -1\}^n$ (i.e., the marginal onto any k subset of coordinates is uniform on $\{1, -1\}^k$.). The goal of this problem is to show that the support size of μ is at least $(cn)^{\lfloor k/2 \rfloor}$ for some constant $c > 0$. Let k be even and suppose $X = (X_1, \dots, X_n)$ is drawn according to μ.

(a) Show that $\mathbb{E}[(X_1 + X_2 + \cdots + X_n)^k] \leq (nek)^{k/2}$. [2 points]

You can do this from first principles by simplifying some binomial sums. For example, you could start by arguing that

$$\mathbb{E}[(X_1 + X_2 + \cdots + X_n)^k] = \sum_{i_1, i_2, \dots, i_n \text{: even; } \sum_j i_j = k} \frac{k!}{(i_1!)(i_2!) \cdots (i_n!)}.$$

(b) Use the above show that the support of μ must be at least $(n/ek)^{k/2}$. [1 point]

5. Let $1 \leq k \leq n$ and $0 \leq \varepsilon \leq 1$. Give an efficient algorithm to generate a random variable X over $\{0,1\}^n$ that is k-wise ε-biased using only $O(\log k + \log \log n + \log(1/\varepsilon))$ random bits. Here, k-wise ε-biased means that for every $I \subseteq [n]$ with $|I| \leq k$, $|\mathsf{Pr}[\oplus_{i \in I} X_i = 0] - 1/2| \leq \varepsilon$. [3 points]
(Hint: Try to combine a "linear" k-wise independent generator, i.e., $Y = yH$ for $y \in_u \{0,1\}^{n-D}$ and $H \in \{0,1\}^{n-D \times n}$ the parity-check matrix of a suitable code, with an ε-biased generator.)