CPS630: Homework 1

Problem 1: Suppose that we independently roll two standard six-sided dice. Let X_1 denote the number on the first dice, X_2 denote the number on the second dice, and let $X = X_1 + X_2$. Calculate the following:

1. $\mathbf{E}[X \mid X_1 \text{ is even}]$.
2. $\mathbf{E}[X \mid X_1 = X_2]$
3. $\mathbf{E}[X_1 \mid X = 9]$
4. $\mathbf{E}[X_1 - X_2 \mid X = k]$ for $k = \{2, 3, \ldots, 12\}$.

Problem 2: In the coupon collector's problem, suppose there are kn coupons, organized into n disjoint sets of k coupons each. Every time step, one of the kn coupons chosen at random is drawn. The experiment stops when at least one coupon from each of the n sets has been collected. What is the expected number of draws till the experiment stops?

Problem 3: Let $a_1, a_2, \ldots, a_n$ be a random permutation of $\{1, 2, \ldots, n\}$, that is, a permutation chosen uniformly at random from all $n!$ permutations.

1. Give a $O(n)$ time algorithm to output such a permutation. Assume a $O(1)$ time subroutine that outputs an integer chosen uniformly at random from $\{1, 2, \ldots, n\}$ whenever it is called.
2. When sorting this permutation, $|a_i - i|$ is the distance a_i needs to move from its current position (i) to its position in the sorted order (a_i). What is

$$\mathbf{E}\left[\sum_{i=1}^{n} |a_i - i|\right]$$

   where the expectation is over the random choice of the permutation? What does the above quantity signify?

3. Hence calculate the expected running time for BubbleSort when the input is a random permutation of the elements. Recall that BubbleSort iteratively swaps two neighboring elements if they are inverted in the sorted ordering.

Problem 4: Consider performing a blood test on a group of n people for a disease. Suppose each person has probability p of testing positive independent of the remaining people.

Consider the following group testing scheme: Divide the n people into $s = n/k$ groups of k people each (assume k divides n so that s is an integer). In each group, pool the blood of all k people and test once; if this test is negative, none of the people have the disease (and we have only tested once); if it returns positive, we need to test all k people separately for the disease, for a total of $k + 1$ tests. What is the expected number of tests performed as a function of n, p, k? For given n and p, what is the optimal value of k? Hence show that there are values of p for which group testing has fewer expected number of tests than n.

Problem 5: An r-way cut is a set of edges whose removal disconnects the graph into at least r connected components. Generalize Karger's contraction algorithm to find a minimum r-way cut, and calculate the probability it succeeds in one iteration.

Problem 6: Given a graph $G(V, E)$, the goal is to find the subgraph $G(V', E')$ such that $|E'|/|V'|$ is maximized. This the DENSEST SUBGRAPH problem. Consider the following linear program for the problem, where there is a variable $y(e)$ for every edge e, and a variable $x(v)$ for every vertex v:

$$\text{Maximize} \sum_{e \in E} y(e)$$

$$\begin{array}{rcll} \sum_{v \in V} x(v) & = & 1 & \\ y(e) & \leq & \min(x(u), x(v)) & \forall e = (u, v) \\ y(e) & \geq & 0 & \forall e \in E \\ x(v) & \geq & 0 & \forall v \in V \end{array}$$

Show that the value of the optimal solution to this linear program is an upper bound to the value of the density of the densest subgraph. Think about how you would make the optimal densest subgraph a feasible solution to the above program.

Consider the following rounding procedure. Choose a number $r \in [0, 1]$, and consider the sets $V_r = \{v \in V | x(v) \geq r\}$, and $E_r = \{e \in E | y(e) \geq r\}$. Show the following:

1. $E_r \subseteq V_r \times V_r$.
2. $\int_{r=0}^{1} |E_r| dr = \sum_{e \in E} y(e)$.
3. $\int_{r=0}^{1} |V_r| dr = \sum_{e \in E} x(v) = 1$.
4. Show that there exists r such that $\frac{|E_r|}{|V_r|} \geq \frac{\sum_{e \in E} y(e)}{\sum_{v \in V} x(v)}$. Therefore show that the optimal solution to the linear programming relaxation actually yields the optimal densest subgraph. In other words, the densest subgraph problem is polynomial time solvable. How would you find what $r \in [0, 1]$ to use for constructing the best V_r and E_r?