CS 787: Advanced Algorithms	**Homework 3**
Out: 3/14/19	**Due: 4/4/19**

Note: Homework should be submitted in pairs on Canvas. We will only accept PDF files.

1. **(Chain length.)** Consider the following process of picking a random subset of $\{1, \cdots, n\}$. At step 1, we pick a uniformly random element in the set. At step $i > 1$, with x_{i-1} denoting the element picked in the previous step, we pick x_i to be a uniformly random element in the set $\{x_{i-1}+1, \cdots, n\}$. The process ends when the element n is picked. Let S denote the random subset of elements picked in this manner. Determine $\mathrm{E}[|S|]$.

2. **(Coupon Collector.)** Your favorite cereal comes with a toy in each box. There are n different toys and each cereal box contains a uniformly random one. Your goal is to collect at least one of each different kind, and the question is how many cereal boxes you would need to buy to accomplish this goal. Let X denote the (random) number of purchases until you have collected at least one toy of each kind. Let $\mu = \mathrm{E}[X]$.

   (a) Compute μ as a function of n.

   Hint: Let X_i denote the (random) number of purchases until you have collected i different kinds of toys, for $i \le n$. Compute $E[X_i - X_{i-1}]$.

   (b) Prove that $\Pr[X > 2\mu]$ is at most $o(1)$. Try to obtain as small a bound as you can.

   Hint: What is the probability that you haven't yet collected a particular toy within 2μ draws?

   (c) Prove that $\Pr[X < \mu/2]$ is at most $o(1)$. Try to obtain as small a bound as you can.

   Hint: Let A_j denote the event that you have collected a toy of type j in $\mu/2$ draws. Observe that the A_j's are "negatively correlated" events. What does this tell you about the probability of their intersection?

3. **(Lower bound for online matching.)** Consider the online bipartite matching problem. For this question, we will allow an online algorithm to come up with a fractional matching, in other words, a feasible solution to the bipartite matching LP. When a vertex in the set R arrives, the online algorithm should determine and commit to a fractional value for all of the edges incident on this vertex without knowing which vertices and edges will arrive in the future. In this problem, you will develop a lower bound on the performance of any fractional online matching algorithm. Observe that the same lower bound should apply to integral randomized matching algorithms as well.

   We will take on the role of an adversary that generates a random bipartite graph independent of the decisions made by the online algorithm. The graph has $2n$ vertices, n in each of L and R. Let us number the vertices in R from 1 through n in order of their arrival. Vertex 1 is connected to all of the vertices in L. Vertex 2 is connected to all but a uniformly random vertex in L. Vertex 3 is connected to all of the neighbors of vertex 2 except for one picked uniformly at random. And so on. Specifically, for $i > 1$, the neighborhood of vertex $i \in R$ consists of all of the neighbors of vertex $i-1$ except for one chosen uniformly at random from this neighborhood.

   One way of thinking about this randomly generated graph is to specify a random permutation π on the n vertices in L. Let π_j denote the jth vertex in this permutation. Then, $i \in R$ is connected to the vertices $\pi_i, \pi_{i+1}, \cdots, \pi_n$ in L.

   Observe that this graph contains a matching of size n regardless of the permutation π: $\{(i, \pi_i)\}$ is such a matching. Therefore, OPT $= n$. We will now show that any online fractional matching is of small size. Let x denote the fractional matching produced by the online algorithm.

   (a) For $i, j \in [n]$, determine $\mathrm{E}[x_{\pi_j,i}]$ where the expectation is taken over the choice of π.

   (b) Determine an upper bound on $\sum_{i,j\in[n]} \mathrm{E}[x_{\pi_j,i}]$. To obtain full points, your upper bound should be close to $(1-1/e)n$.

4. **(Bin-Packing.)** The **online bin-packing** problem is a variant of the knapsack problem. We are given an unlimited number of bins, each of size 1. We get a sequence of items one by one, each of a certain size no more

than 1, and are required to place them into bins as we receive them. Our goal is to minimize the number of bins we use, subject to the constraint that no bin to should be filled to more than its capacity. In this question we will consider a simple online algorithm for this problem called **First-Fit** (FF). FF orders the bins arbitrarily, and places each item into the first bin that has enough space to hold the item.

Recall that the competitive ratio of an online algorithm for a minimization problem is the maximum over all arrival sequences of the cost of the algorithm (in this case, the number of bins used by the algorithm) to the cost of the hindsight OPT (in this case, the minimum number of bins necessary to pack the items).

(a) Give an instance of bin-packing for which FF does not obtain the optimal packing.

(b) Prove that FF has competitive ratio no more than 2, that is, on every instance, it uses no more than twice as many bins as necessary. Can you prove a better bound than 2?