CS3230: Design and Analysis of Algorithms
Semester 2, 2021-22, School of Computing, NUS

Practice Problem Set: Probabilistic Techniques

February 17, 2022

Instructions

- This problem set is **completely optional**. There is no need to submit the solutions.
- Solutions will be available later. You are strongly encouraged to first try to solve the problems by yourselves and then check the solutions.
- Post on the LumiNUS forums if you will face any problem while solving the questions.

Question 1: Consider an undirected graph $G = (V, E)$ with $|V| = n$ and $|E| = m$. Order the vertices in V randomly. Let I be the set of vertices v whose all the neighbors occur after v in the above random order. Note, I is an independent set, i.e., no edges between the vertices in I (think why?). Show that expected size of I is $\sum_{v \in V} \frac{1}{d(v)+1}$, where $d(v)$ denotes the degree of v in G.

Question 2: In a fictional dice game, you repeatedly toss a fair (i.e., six sides are equally likely) die. If it ever turns to be an odd number, you lose. Otherwise, your score is the number of tosses that it takes you to get the first 6. What is the probability that your score is 1 conditioned on the event that you do not lose.

Question 3: Given a permutation π over $\{1, 2, \cdots, n\}$, let $L(\pi)$ be the length of the longest increasing subsequence in π. (Note, a subsequence may not be contiguous. E.g., in the permutation $\pi = (1, 6, 4, 5, 2, 7, 3)$, the longest increasing subsequence is $(1, 4, 5, 7)$ and thus $L(\pi) = 4$.)

Show that for a random permutation π over $\{1, 2, \cdots, n\}$, $\mathbb{E}[L(\pi)] = O(\sqrt{n})$. (Additionally, also try to show that $\mathbb{E}[L(\pi)] = \Omega(\sqrt{n})$.)

Question 4: Suppose n people queue up in a movie theatre which has exactly n seats. However, the first person in the queue has lost his ticket and sits in one of the empty seats uniformly at random. Subsequently, each person (everybody else has their tickets) sits either in his assigned seat, or if that seat is already occupied, seats in an empty seat uniformly at random. What is the expected number of people not sitting in their assigned seats?

Question 5: Consider a connected graph $G = (V, E)$ with n nodes (i.e., $|V| = n$) and m edges (i.e., $|E| = m$). For any (non-trivial) partitioning of $V = V_1 \cup V_2$ (with $V_1 \cap V_2 = \emptyset$ and both V_1 and V_2 are non-empty) we call the set of all the edges having one endpoint in V_1 and other in V_2 a *cutset*. Now we would like to find a partition such that the corresponding cutset is of minimum size.

To find such a partition we would adopt the following randomized procedure (Procedure $\mathcal{A}$):

1. Pick an edge of G uniformly at random, and contract its two endpoints into a single node. Repeat this process until only two nodes remain.
2. Let u_1, u_2 be the two remaining nodes at the end. Let V_1 be the set of nodes that went into u_1 (while contracting edges), and similarly V_2 be the set of nodes that went into u_2. Output V_1 and V_2 as the partition.

Just to clarify, suppose the algorithm choose an edge $u - v$ to contract, then you may think of the new node as a set uv and all the edges (except $u - v$) incident on u and v in original will now incident on this new node. Note, contraction of edges may create multiple edges between two nodes. See Figure 1. Now answer the following questions.

(a) Show that the size of a minimum cutset is at most $\frac{2m}{n}$.

(b) Probability that the cutset corresponding to the output (partition) of Procedure $\mathcal{A}$ is of minimum size, is at least $\frac{2}{n(n-1)}$.

(c) Repeat Procedure $\mathcal{A}$ for k times, and finally return the partition with minimum cutset size among k output partitions. Now this is your final algorithm. What should be the value k you will choose such that your final algorithm outputs the partition with minimum cutset, with probability at least 2/3? (Provide the detailed analysis.)

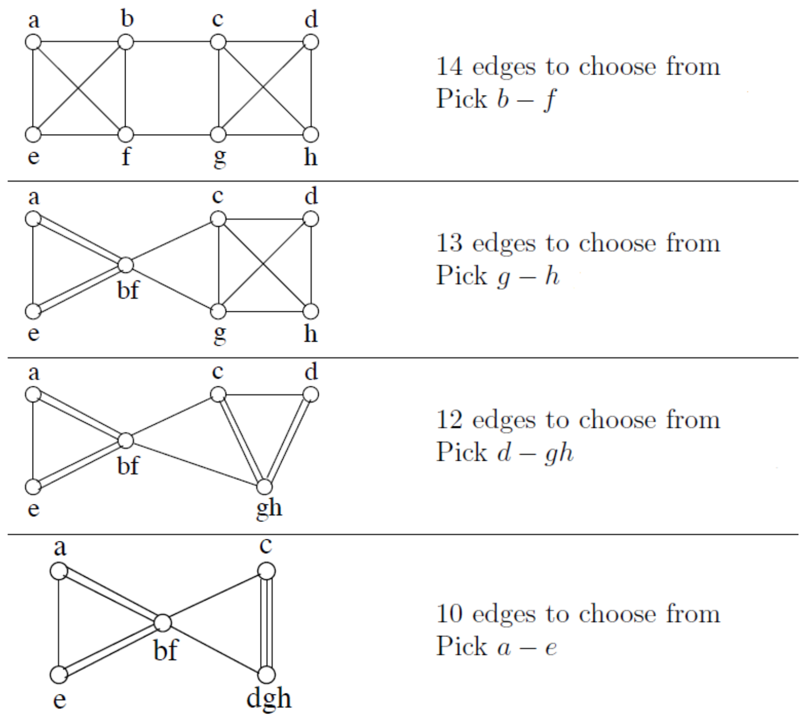

Figure 1: An example run of the algorithm