**CS 466/666: Algorithm Design and Analysis** **University of Waterloo: Fall 2024**

# Homework 2

Due: Thursday, October 10, 2024

**Problem 1.** We reexamine the balls-and-bins experiment in this question, focusing on aspects other than the maximum load. Suppose we are throwing a number of balls into $n$ bins one by one where each ball chooses one of the bins independently and uniformly at random.

($a$) Prove that the expected number of balls we need to throw before every bin has at least one ball inside it is $\Theta(n \log n)$. **(10 points)**

($b$) Find the sufficient and necessary asymptotic value for the number of balls we need to throw, so that with constant probability, at least one bin has two or more balls inside it. **(10 points)**

**Problem 2.** Recall that a cut in an undirected graph $G = (V, E)$ is any partition of vertices $(S, \bar{S})$ of $V$ and $\delta(S)$ denotes the set of edges in the cut $S$, i.e., the edges between $S$ and $\bar{S}$. In the **minimum cut** problem, we are interested in finding a cut $(S, \bar{S})$ with the minimum number of cut edges, i.e., minimize $|\delta(S)|$.

We examine a by-now classical algorithm for this problem due to Karger.

($a$) Suppose $S$ is a minimum cut of $G$ and $e = (u, v)$ is <u>not</u> a cut edge of $S$. Prove that if we contract $u$ and $v$ into a single vertex and <u>keep</u> the parallel edges (but remove self-loops), then, the minimum cut size of $G$ remains the same after the contraction. **(5 points)**

($b$) Fix some minimum cut $(S, \bar{S})$ in $G$. Prove that if we sample an edge uniformly at random from $G$, then, the probability that $e$ is a cut edge of $S$ is at most $2/n$. **(10 points)**

*Hint:* What is the relationship between the number of edges, the minimum degree, and the minimum cut size in a graph?

($c$) Consider the following algorithm:

> **Algorithm 1.** Given an undirected multi-graph $G = (V, E)$,
>
> ($i$) Sample an edge $e = (u, v)$ uniformly at random from $G$;
>
> ($ii$) Contract the set $\{u, v\}$ in $G$ to obtain $G_{\{u,v\}}$;
>
> ($iii$) Recurse on $G_{\{u,v\}}$ and continue until only two vertices are remained; then, return the sets corresponding to these two (possibly) contracted vertices.

Prove that this algorithm outputs a minimum cut of a given graph $G$ with probability at least $\binom{n}{2}^{-1}$. **(10 points)**

($d$) Use the above algorithm to prove that in every graph $G$, there can be at most $\binom{n}{2}$ minimum cuts. **(5 points)**

**Problem 3.** Let $G = (V, E)$ be an $n$-vertex $m$-edge undirected graph with weights $w(e)$ for each edge $e \in E$. Suppose $T \subseteq E$ is some spanning tree of $G$ (not necessarily an MST). Design a deterministic $O(m \log n)$ time algorithm that outputs all $T$-heavy edges of $G$. Recall that an edge $e \in E$ is $T$-heavy if $e$ is not in $T$ and in the cycle created by adding $e$ to $T$, $e$ has the maximum weight. **(20 points)**

**Problem 4.** Let $G = (V, E)$ be an undirected graph with weights $w(e)$ for each edge $e \in E$. An **edge cover** of $G$ is a set of edges $F \subseteq E$ in $G$ such that every vertex is incident on at least one edge in $F$.

($a$) Write a LP relaxation for finding a minimum weight edge cover of a given graph. **(15 points)**

($b$) Suppose maximum degree of $G$ is some given $\Delta \geqslant 1$. Let $x \in \mathbb{R}^E$ be an optimal solution to your LP from the previous part. Show that if we pick each edge $e$ with $x_e \geqslant 1/\Delta$ we obtain a feasible edge cover of $G$ with weight at most $\Delta$ times the minimum weight edge cover of $G$. **(15 points)**

**Problem 5 (Extra Credit).** Design a *deterministic* $O(m)$ time algorithm for finding MST of a given *planar* graph. A planar graph is a graph that can be drawn on a surface so that no two edges cross each other.

**(+10 points)**

*Hint:* This is a pretty simple question. Think about (or search) how many edges are in a planar graph? And, what happens if you contract an edge in a planar graph?

**Problem 6.** Let $G = (V, E)$ be a graph on $n$ vertices obtained by adding an edge between each pair of vertices independently and with probability

$$p := \frac{100 \log n}{n}.$$

Prove that with high probability the **chromatic number** of $G$ is $\Omega(\frac{\log n}{\log \log n})$. Recall that the chromatic number is the minimum number of colors we can assign to the vertices so that no edge receives the same color on both its endpoints. **(+10 points)**

**Problem 7.** Design a deterministic algorithm for Problem 3 with $O(m \log \log n)$ runtime (or even faster).

**(+10 points)**