

## **Precept Outline**

- · Analysis of Karger's Algorithm
- · Approximate Counting

## A. Analysis of Karger's Algorithm

In lecture, we state but don't prove a  $\Omega(1/V^2)$  lower bound for one iteration of Karger's algorithm; the goal of this problem is to prove this fact.

Fix a mincut  $C \subset E$  in the graph G; we will prove something slightly stronger, in fact: that *no edge* in C is added to the forest throughout the execution of Kruskal's algorithm with random edge weights.

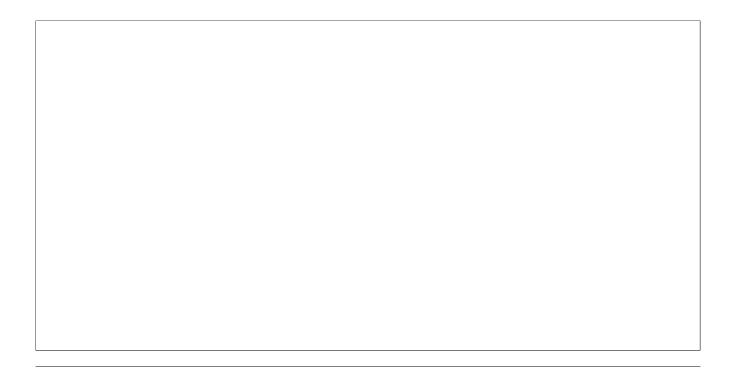
Start by proving that |C| is a lower bound on the *minimum degree*  $\delta(G) \coloneqq \min_{v \le V} \{d(v)\}$  of G.

Next, prove that  $E \ge |C|V/2$  and that the first edge added by Kruska's algorithm is in C with probability  $\le 2/V$ .

Finally, define  $p_{n,k}$  as the probability that (a fresh run of) Karger's algorithm on an n-vertex graph never adds an edge from a fixed set of k edges. Prove the recurrence

$$p_{n,k} \ge \left(1 - \frac{2}{n}\right) p_{n-1,k}$$

and conclude that one iteration of Karger's algorithm succeeds with probability  $\geq \frac{2}{V(V-1)} \sim \frac{2}{V^2}$ .



## **B.** Approximate Counting

In this exercise, we'll learn how to use randomness for a somewhat unintuitive purpose: to save space. Recall that we need  $\log_2 n$  bits to store a counter with value  $0 \le v < n$ ; our goal is to store an *approximate* counter—trading space for accuracy in the sense explained next—with only  $\Theta(\log\log n)$  bits.

The approximate counting algorithm is the following:

- Initialize a variable k := 0.
- At every event, increment k with probability  $1/2^k$  (and leave k unchanged with probability  $1-1/2^k$ ).

Let  $0 \le C_n \le n$  be the random variable corresponding to the value of the counter after n events. Prove that  $\mathbb{E}[2^{C_n}] = n+1$  for all  $n \in \mathbb{N}$ . Hint: find a recurrence for  $\mathbb{P}[C_n = k]$ .

