0.1 Global Min-Cut

We start by stating the question we are trying to solve.

Question: Global Min-Cut Problem

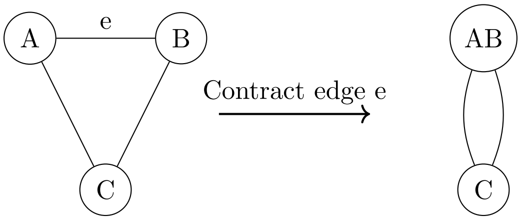
We are given an unweighted and undirected graph $G = (V, E)$. A "cut" of the graph is a partition of the vertices into two sets such that each set is non-empty. The "cost" of the cut (also called the cut "size") is the number of edges that have one endpoint in each set. The goal is to find the cut with the smallest cost.

Let $n := |V|$. We consider two randomized algorithms:

- Karger's Algorithm (1995) - poly(n)-time.
- Karger-Stein Algorithm (1996) - $\tilde{O}(n^2)$ time.

0.1.1 Edge Contraction

The key idea in both algorithms is edge contraction. Given an edge (u, v), we can contract it by removing the edge and merging the two vertices into a single vertex. The new vertex will have all the edges that u and v had. For example:



Each contraction reduces the number of vertices by 1, and we want to stop when we have two vertices left. This leaves us with two vertices and many edges between them.

Note:
We allow multi-edges.

Question: Should we keep self-edges?
Answer: No. Delete them.
The observation we make is that we win iff edges at the end are a min-cut.

0.1.2 Karger's Algorithm

The algorithm is as follows:

- Randomly pick an edge (u, v).
- Contract (u, v).
- Repeat until 2 vertices are left.

Lenma 0.1.1
For any min-cut C, the probability that Karger's algorithm finds C is $\geqslant \frac{1}{\text{poly}(n)}$.

Proof. Let $e_1, e_2, \ldots$ be the edges we contract. Let's start by considering $P(e_1 \in C)$.

We first have $P(e_1 \in C) = \frac{|C|}{|E|}$, but there's something better to notice. Let d be the minimum degree of any vertex in G. Then $|C| \leqslant d$ and $|E| \geqslant \frac{dn}{2}$. So $P(e_1 \in C) \leqslant \frac{2}{n}$ and $P(e_1 \notin C) \geqslant 1 - \frac{2}{n}$.

Now consider $P(e_2 \notin C \mid e_1 \notin C)$. By the same analysis, we have $P(e_2 \notin C \mid e_1 \notin C) \geqslant 1 - \frac{2}{n-1}$. This is because after we contract e_1, we have $n-1$ vertices left. So, C is still a min-cut with respect to the remaining vertices. Set d as the new minimum degree, and we have $|C| \leqslant d$ and $|\text{remaining edges}| \geqslant \frac{d(n-1)}{2}$. Then:

$$P(e_2 \in C \mid e_1 \notin C) = \frac{|C|}{|\text{remaining edges}|} \leqslant \frac{d}{\frac{d(n-1)}{2}} = \frac{2}{n-1}.$$

Now:

$$\begin{gathered} P(e_3 \notin C | e_1, e_2 \notin C) \geqslant 1 - \frac{2}{n-2} \\ \vdots \\ P(e_k \notin C | e_1, \dots, e_{k-1} \notin C) \geqslant 1 - \frac{2}{n-k+1}. \end{gathered}$$

So,

$$\begin{aligned} P(\text{we find } C) = P(e_1, \dots, e_{n-2} \notin C) &\geqslant \left(\frac{n-2}{n}\right)\left(\frac{n-3}{n-1}\right)\cdots\left(\frac{1}{3}\right) \\ &\approx \frac{2}{n(n-1)}. \end{aligned}$$

☺

Conclusion: The algorithm succeeds with probability $\geqslant \frac{2}{n(n-1)}$.
New Goal: Find min-cut with $\geqslant 1 - \frac{1}{n^2}$ probability.
Idea: Repeat t times and return the best cut we find. (We will determine the value for t below.)

The probability that all t trials fail is $\leqslant \left(1 - \frac{2}{n(n-1)}\right)^t$. We want this to be $\leqslant \frac{1}{n^2}$. We utilize the inequality below:

$$\left(1 - \frac{1}{k}\right)^k \leqslant \frac{1}{e} \leqslant \left(1 - \frac{1}{k}\right)^{k-1}.$$

So,

$$\begin{aligned} \left(1 - \frac{2}{n(n-1)}\right)^t &\leqslant \frac{1}{e^{t \cdot \frac{2}{n(n-1)}}} \implies t = 2\binom{n}{2}\log n \\ &\leqslant \frac{1}{e^{2\log n}} = \frac{1}{n^2}, \end{aligned}$$

as desired.

0.1.3 Karger-Stein Algorithm

We start with the following motivation. We have:

$$P(\text{first } n/2 \text{ contractions succeed}) \geqslant \frac{n-2}{n}\frac{n-3}{n-1}\cdots\frac{n/2-1}{n/2+1} \approx \frac{1}{4}.$$

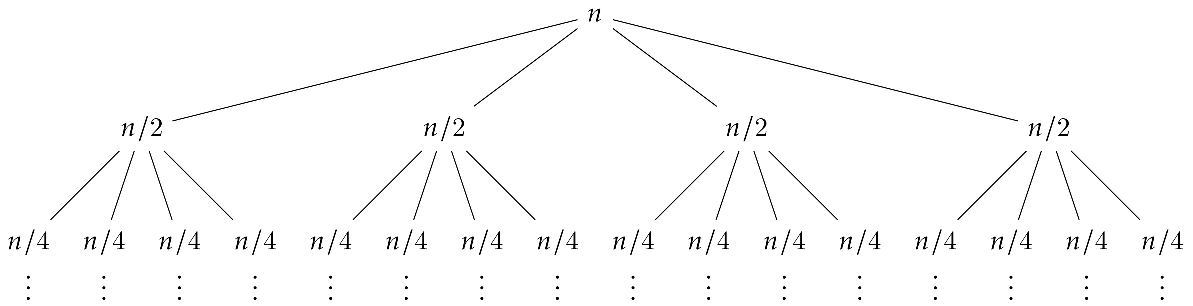
This is because after we telescope, the bottom two terms are both near n while the top two terms are near $n/2$. So now, if the first $n/2$ contractions succeed, what's the probability that the next $n/4$ succeed?

$$P(\text{next } n/4 \text{ contractions succeed}) \geqslant \frac{n/2-2}{n/2}\frac{n/2-3}{n/2-1}\cdots\frac{n/4-1}{n/4+1} \approx \frac{1}{4}.$$

And this pattern continues.

We now go over the Karger-Stein algorithm:

- Represent the problem as a tree.
- For the first $n/2$ contractions, run Karger's algorithm 4 times.
- For the next $n/4$ contractions, run Karger's algorithm 16 times.
- So on, until the base case of the algorithm which is when there are two vertices left.



Time Analysis: The first level takes $\tilde{O}(n^2)$ time. The second level takes $\tilde{O}\left(\frac{n^2}{2}\right)$ time. The third level takes $\tilde{O}\left(\frac{n^2}{4}\right)$ time. So the total time is $\tilde{O}(n^2)$.
So, at level i:

- There are 4^i edges.
- The work per edge $\leqslant (|\text{remaining vertices}|)^2 = \tilde{O}\left(\frac{n}{2^i}\right)^2 = \tilde{O}\left(\frac{n^2}{4^i}\right)$.

This implies that the cost of the ith level is less than $\tilde{O}(n^2)$. Summing, we get a total cost $\tilde{O}(n^2)$.

Now we consider the probability that our algorithm succeeds. Again, we have a 4-ary tree with a depth of $t = O(\log n)$. Each edge flips a coin where $P(\text{heads}) = \frac{1}{4}$. So what we want is a t length chain of all heads. We can show the following result:

$$P(\text{good path}) \geqslant \frac{1}{t} \geqslant \Omega\left(\frac{1}{\log n}\right).$$

The goal is an algorithm that succeeds with probability greater than $1 - \frac{1}{n^2}$.
Exercise: $O(\log^2 n)$ repetitions suffice.