1 Minimum cut

1.1 Terminology

1. **Multigraph**: A graph allowed to have more than one edge between two vertices.
2. **Cut**: A set of edges whose removal disconnects the graph.
3. **Size of a cut**: Number of edges in the cut.

1.2 Deterministic algorithm

1. Pick a vertex $s \in V$.
2. For each remaining vertex $t \in V - \{S\}$, find a minimum $s - t$ cut.
3. Return the minimum size cut over all cuts computed.

1.3 Karger's algorithm

1.3.1 Edge contraction

Let $e = \{u, v\} \in E$, then $G - e$ is the result of contracting vertices u and v into a new vertex w. Keep all multi-edges and remove all self loops.

1.3.2 Implementation

1. Select an edge e uniformly at random and contract it.
2. Return the cut of the original graph corresponding to the cut between the last two remaining edges.

1.3.3 Probabilistic analysis

Suppose S is a minimum cut of G.(Worst case: S is unique.), then the algorithm is successful $\iff$ all edges in S are not contracted until termination. Let ε_i be the event that the ith sampled edge does not belong to S, then $\mathbf{P}(\text{Karger's is successful}) \geq \mathbf{P}(\text{Karger's returned } S) = \mathbf{P}(\varepsilon_1 \cap \cdots \cap \varepsilon_{n-2})$.
Let δ be the minimum degree of the graph, then $|E| \geq \frac{\delta n}{2}$ and $|S| \leq \delta$. $\mathbf{P}(\varepsilon_1) = 1 - \frac{|S|}{|E|} \geq 1 - \frac{2}{n}$. Similarly, we have $\mathbf{P}(\varepsilon_2|\varepsilon_1) \geq 1 - \frac{2}{n-1}$. In general, we have $\mathbf{P}(\varepsilon_i|\varepsilon_1 \cap \dots \varepsilon_{i-1}) \geq 1 - \frac{2}{n-i+1}$, thus by properties of conditional probability, we have $\mathbf{P}(\varepsilon_1 \cap \cdots \cap \varepsilon_{n-2}) \geq \frac{2}{n(n-1)}$
We have a corollary that the number of minimum cuts is at most $\frac{n(n-1)}{2}$.

1.3.4 Technique: Repeat to improve polynomial small success rate

Note that if the success rate is polynomial small, we can often amplify the success probability by repetition of polynomial times. Let $\varepsilon = \frac{2}{n(n-1)}$. We claim that the success probability can be amplified to $1 - f$ by repeating for $t = \lceil \varepsilon^{-1} \ln f^{-1} \rceil \in O(n^2 \lceil \log f^{-1} \rceil)$, and the return a cut of the smallest size, among these cuts.
The failure probability(i.e.no successful execution) is at most

$$(1 - \varepsilon)^t \leq e^{-\varepsilon t} \leq f$$

Therefore, the success probability is at least $1 - f$.

1.3.5 Time complexity

Each iteration takes $O(n^2)$. Repetition will take $O(n^4 \lceil \log f^{-1} \rceil)$.

2 Matrix multiplication

2.1 Terminology

Given $A, b, C \in \mathbb{R}^{n*n}$, verify $AB = C$.

2.2 Freivald's algorithm

2.2.1 Implementation

1. Let $S = \{s_1, \dots, s_k\}$, then $|S| = k$.
2. Let $v = (v_1, \dots, v_n)$, with each entry sampled uniformly and randomly from S.
3. Compute $r = Cv, s = Bv$ and then $t = As$, compare r and t. Note that we do not carry matrix multiplication but use matrix to multiply a vector, which reduces time complexity to $O(n^2)$.

2.2.2 Probabilistic analysis

We focus on the case of $AB \neq C$. We define $C' = AB - C$ and $u = C'v = t - r$. The algorithm fails $\iff$ $u = 0$. As long as u has a nonzero entry, the algorithm is successful. We can fix an entry $c'_{ij} \neq 0$. Note that $u_i = K + c'_{ij} v_j$, where K is the rest of the dot product. By principle of deferred decision, once all entries except v_j are revealed, there is at most one choice of v_j to make $u_i = 0$. Hence $\mathbf{P}(u_i = 0) \leq \frac{1}{k}$. $\mathbf{P}(\text{Algorithm is successful}) = \mathbf{P}(u \text{ has nonzero entry}) \geq \mathbf{P}(u_i \neq 0) \geq 1 - \frac{1}{k}$. By choosing $k = 100$, we guarantee success probability of at least 0.99.

3 Polynomial identity testing

3.1 Terminology

Given polynomial $P(x_1, \dots, x_n)$ with degree d, which could be prohibitively expensive to compute for expansion, check whether P is zero polynomial.

3.2 Implementation

1. Let S be any set of numbers. $k = |S|$.
2. Sample each x_i uniformly at random from S.
3. If $P(x_1, \dots, x_n) = 0$, we decide that P is zero polynomial.

3.3 Probabilistic analysis

We focus on the case $P \neq 0$. The algorithm fails $\iff$ $P(x_1, \dots, x_n) = 0$. We prove by induction on number of variables that this probability is at most $\frac{d}{k}$.
Base case: P has one variable x_1, then P has at most d real roots by fundamental theorem of algebra, which has a probability of at most $\frac{d}{k}$ to be selected.
Inductive step: We consider $n > 1$, suppose the probability holds for smaller n. We write out canonical expansion with respect to x_1:

$$P(x_1, \dots, x_n) = \sum_{i=0}^{d} x_1^i P_i(x_2, \dots, x_n)$$

We consider the leading term $x_1^q P_q(x_2, \dots, x_n)$
Let ϵ denote the event that $P_q(x_2, \dots, x_n) = 0$, then

$$\begin{aligned}\mathbf{P}(P(x_1, \dots, x_n) = 0) &= \mathbf{P}(P(x_1, \dots, x_n) = 0|\epsilon)\mathbf{P}(\epsilon) \\ &+ \mathbf{P}(P(x_1, \dots, x_n) = 0|\neg\epsilon)\mathbf{P}(\neg\epsilon) \\ &< \mathbf{P}(\epsilon) + \mathbf{P}(P(x_1, \dots, x_n) = 0|\neg\epsilon)\end{aligned}$$

Note that $\mathbf{P}(\epsilon) \leq \frac{d-q}{k}$. If ϵ does not occur, we could see P has a power q polynomial of x_1, then by fundamental theorem of algebra we have at most q real roots, thus $\mathbf{P}(P(x_1, \dots, x_n) = 0|\neg\epsilon) \leq \frac{q}{k}$, thus sum is no more than $\frac{d}{k}$, we conclude the proof.

3.4 Perfect matching

Consider a bipartite graph, a perfect matching only exists if there is a permutation such that the edge set contains all edges defined by that permutation. We could check for perfect matching by constructing a square matrix X with $X_{ij} = x$ if there is an edge. We can compute determinant in polynomial time. If a perfect matching is present, this determinant should not be zero polynomial.

4 Coupon collector

4.1 Terminology

Once you buy a box, you have equal probability to obtain one of n coupons, how many boxes should you buy to collect all coupons?

4.2 Construct random variable

Let $X = \sum_{i=1}^{n} X_i$, where X_i is the number of additional boxes to buy to obtain a new coupon, given that you have $i - 1$ coupons. By this formulation, X is the random variable denoting the total number of boxes. By linearity of expectation, we have $\mathbb{E}[X] = \sum_{i=1}^{n} \mathbb{E}[X_i]$.
Note that once we have $i - 1$ coupons, to get a new coupon is to get one from remaining $n - i - 1$ coupons, by uniform probability, the success probability is $\frac{n-i-1}{n}$, and repeat until one success, the number of attempts is clearly a geometric distribution with parameter $\frac{n-i-1}{n}$, and expectation is $\frac{n}{n-i-1}$. We evaluate the sum

$$\begin{aligned}\mathbb{E}[X] &= \sum_{i=1}^{n} \mathbb{E}[X_i] \\ &= \sum_{i=1}^{n} \frac{n}{n-i+1} \\ &= n \sum_{i=1}^{n} \frac{1}{n-i+1} \\ &= n \sum_{j=1}^{n} \frac{1}{j} \\ &\in O(n \log n)\end{aligned}$$

5 Concentration inequalities

5.1 Markov inequality

5.1.1 Formulation

If X is a non-negative random variable and $a > 0$, then

$$\mathbf{P}(X \geq a\mathbb{E}[X]) \leq \frac{1}{a}$$

5.1.2 Proof

$$\begin{aligned}\mathbb{E}[X] &= \sum_x x\mathbf{P}(X=x)\\ &\geq \sum_{x\geq a\mathbb{E}[X]} a\mathbb{E}[X]\mathbf{P}(X=x)\\ &= a\mathbb{E}[X]\sum_{x\geq a\mathbb{E}[X]}\mathbf{P}(X=x)\\ &= a\mathbb{E}[X]\mathbf{P}(X\geq a\mathbb{E}[X])\end{aligned}$$

Once we have established Markov inequality, we could realize that small expectation guarantees that probability mass is concentrated around expectation. Suppose $\mathbb{E}[X]\in o(1)$, then take $a=\frac{1}{\mathbb{E}[X]}$, we have $\mathbf{P}(X\geq 1)\leq \mathbb{E}[X]\in o(1)$, which implies that it is unlikely for X to take large values.
However, it does not guarantee such a conclusion if $\mathbb{E}[X]\in\omega(1)$. Suppose we have a random variable Y such that $\mathbf{P}(Y=a^2)=\frac{1}{a}$, and $\mathbf{P}(Y=0)=1-\frac{1}{a}$, by taking arbitrarily large a, we could make $\mathbb{E}[Y]$ arbitrarily large as well, but probability mass is concentrated around 0.

5.2 Chebyshev inequality

5.2.1 Formulation

For a random variable X, we have

$$\mathbf{P}(|X-\mathbb{E}[X]|\geq c)\leq\frac{\mathbf{Var}[X]}{c^2}$$

5.2.2 Proof

For any random variable, we know that $\mathbf{Var}[X]=\mathbb{E}[(X-\mathbb{E}[X])^2]$.
Since $(X-\mathbb{E}[X])^2\geq 0$, we can apply Markov inequality to get

$$\mathbf{P}((X-\mathbb{E}[X])^2)\geq a\mathbb{E}[(X-\mathbb{E}[X])^2])\leq\frac{1}{a}$$

It simplifies to

$$\mathbf{P}(|X-\mathbb{E}[X]|\geq c)\leq\frac{\mathbf{Var}[X]}{c^2}$$

5.3 Chernoff bounds

5.3.1 Formulation

Consider $X\sum_{i=1}^n X_i$, a sum of independent Poisson trials. Let $\mu=\mathbb{E}[X]$, we have several bounds

$$\begin{aligned}&\mathbf{P}(X\geq(1+\delta)\mu)\leq(\frac{e^{\delta}}{(1+\delta)^{1+\delta}})^{\mu}\\ &\mathbf{P}(X\leq(1-\delta)\mu)\leq(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}})^{\mu}\\ &\mathbf{P}(X-\mu\geq\delta\mu)\in e^{-\Omega(\mu\delta\log\delta)},\delta>1\\ &\mathbf{P}(|X-\mu|\geq\delta\mu)\in 2e^{-\Omega(\delta^2\mu)},\delta\in[0,1]\\ &\mathbf{P}(X\geq(1+\delta)\mu)\leq e^{-\frac{\delta^2\mu}{3}},\delta\in[0,1]\\ &\mathbf{P}(X\geq(1+\delta)\mu)\leq e^{-\frac{\delta^2\mu}{2+\delta}},\delta>0\\ &\mathbf{P}(X\leq(1-\delta)\mu)\leq e^{-\frac{\delta^2\mu}{2}},\delta\in[0,1]\end{aligned}$$

5.3.2 Proof

Apply Markov inequality to e^{tX}.

5.3.3 Hoeffding bound

We could generalize Chernoff bound to deal with random variables $X_i\in[a_i,b_i]$.

$$\begin{aligned}&\mathbf{P}(X\leq\mu-t)\leq e^{-\frac{2t^2}{\sum_{i=1}^n(b_i-a_i)^2}}\\ &\mathbf{P}(X\geq\mu+t)\leq e^{-\frac{2t^2}{\sum_{i=1}^n(b_i-a_i)^2}}\end{aligned}$$

6 Supplementary formula

1. $1+x\leq e^x$ for all real x.
2. $1-x\geq e^{\frac{-x}{1-x}}$ if $x<1$.
3. If a given algorithm has a success probability of at least ϵ per execution, we could repeat the algorithm for $t=\lceil\epsilon^{-1}\ln f^{-1}\rceil$ times to obtain a successful execution with probability of at least $1-f$.
4. Fundamental theorem of algebra
5. Harmonic series, $\sum_{i=1}^n\frac{1}{i}\in O(\log n)$
6. Law of total probability: $P(A)=\sum_{i=1}^n P(A\cap B_i)=\sum_{i=1}^n P(A|B_i)P(B_i)$
7. Principle of deferred decision: $P(\epsilon|x=a)\leq p\ \forall a\in A\implies P(\epsilon)=\sum_{a\in A}P(\epsilon|x=a)P(x=a)\leq p\sum_{a\in A}P(x=a)=p$
8. $\mathbf{Var}[X]=E[|X-\mathbb{E}[X]|^2]=\mathbb{E}[X^2]-\mathbb{E}[X]^2\implies\mathbb{E}[X^2]\geq\mathbb{E}[X]^2$
9. $\mathbf{Var}[\sum_{X_i}]=\sum_i\mathbf{Var}[X_i]+\sum_{j<k}2\mathbf{cov}[X_j,X_k]$
10. Cauchy-Schwarz inequality: $|<u,v>|\leq\|u\|\|v\|$

7 Identities for computational complexity

1. $\lim_{n\to+\infty}\dfrac{f(n)}{g(n)}=0\implies f(n)\in o(g(n))$
2. $\lim_{n\to+\infty}\dfrac{f(n)}{g(n)}<+\infty\implies f(n)\in O(g(n))$
3. $0<\lim_{n\to+\infty}\dfrac{f(n)}{g(n)}<+\infty\implies f(n)\in\Theta(g(n))$
4. $\lim_{n\to+\infty}\dfrac{f(n)}{g(n)}>0\implies f(n)\in\Omega(g(n))$
5. $\lim_{n\to+\infty}\dfrac{f(n)}{g(n)}=+\infty\implies f(n)\in\omega(g(n))$
6. $f(n)\in O(g(n))\iff g(n)\in\Omega(f(n))$
7. $f(n)\in o(g(n))\iff g(n)\in\omega(f(n))$