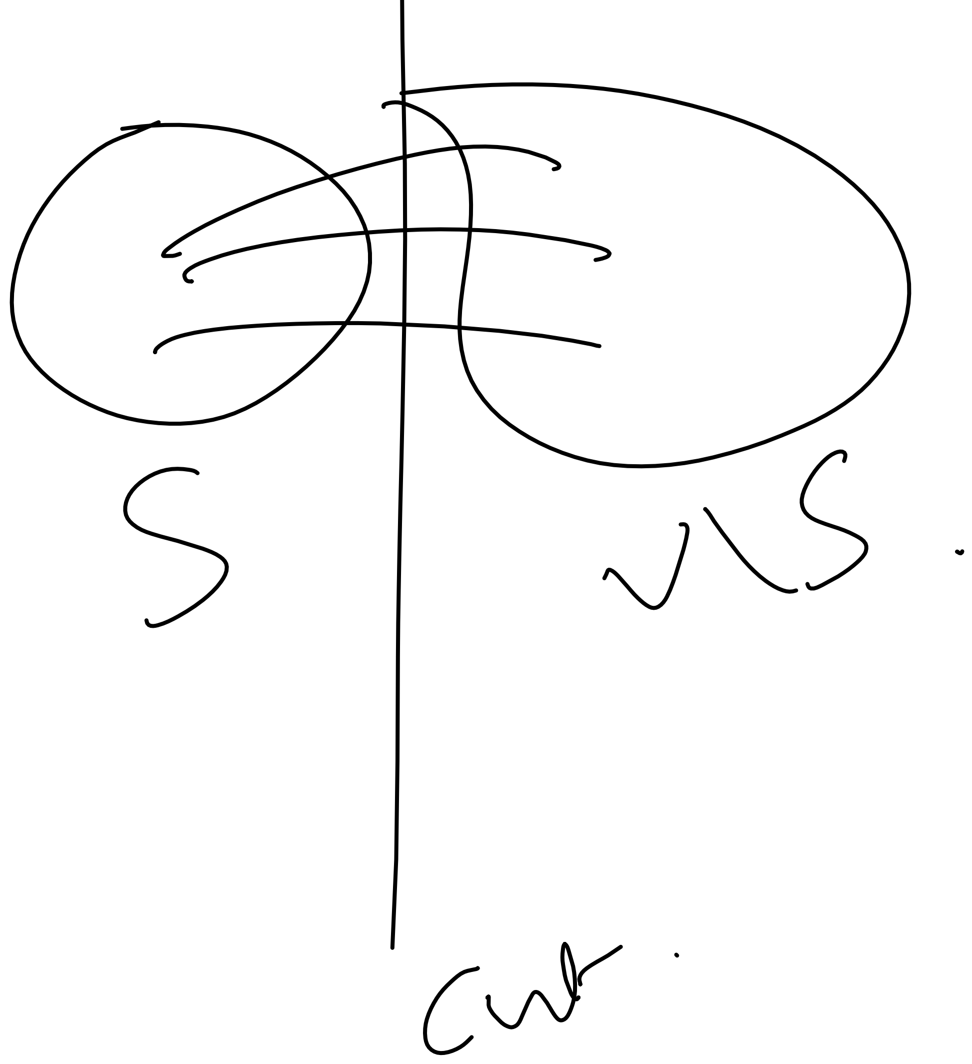
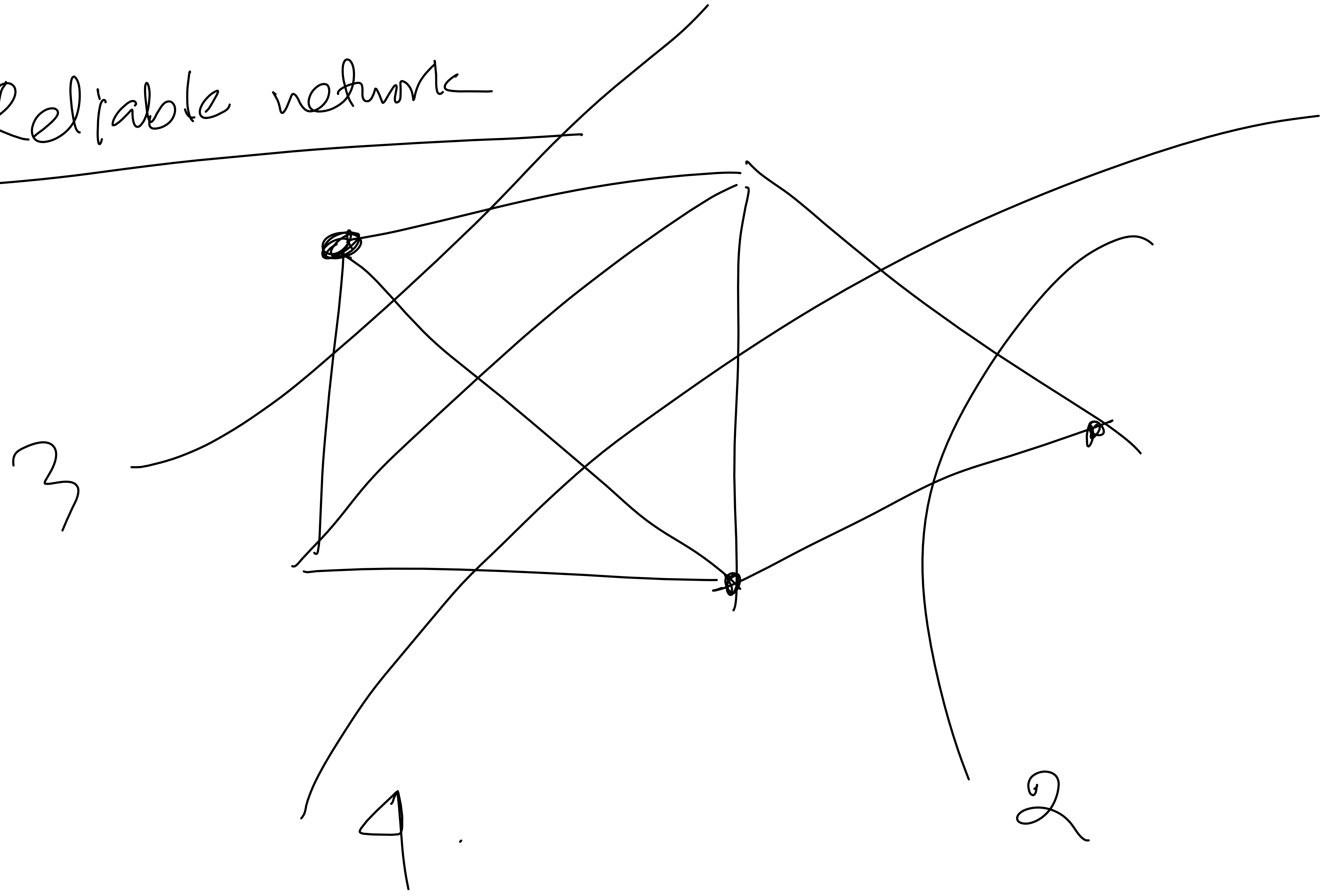
16.10.2024

Min-Cut | Undirected Graph

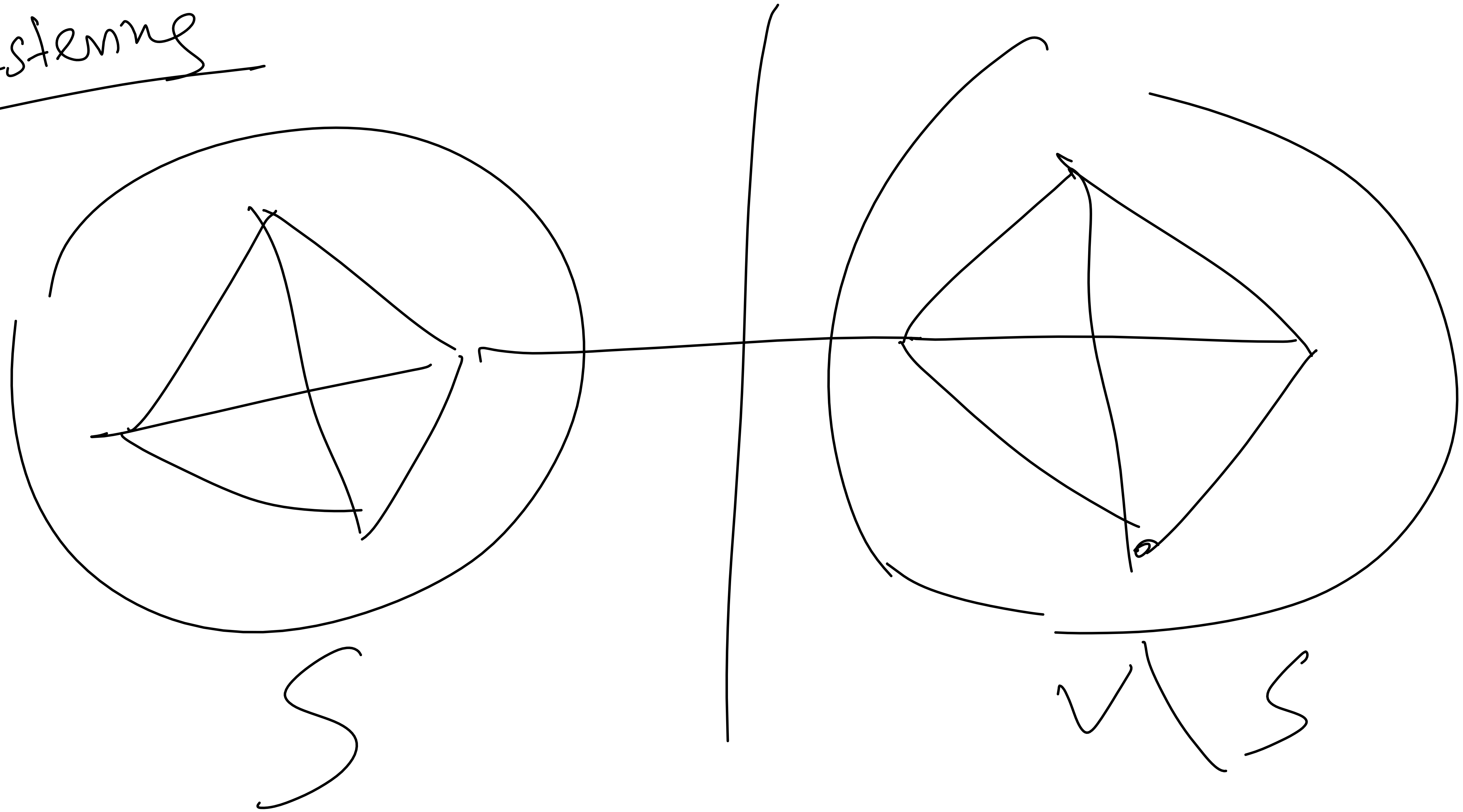


find S : such that edge crossings are as small as possible.

① Reliable network



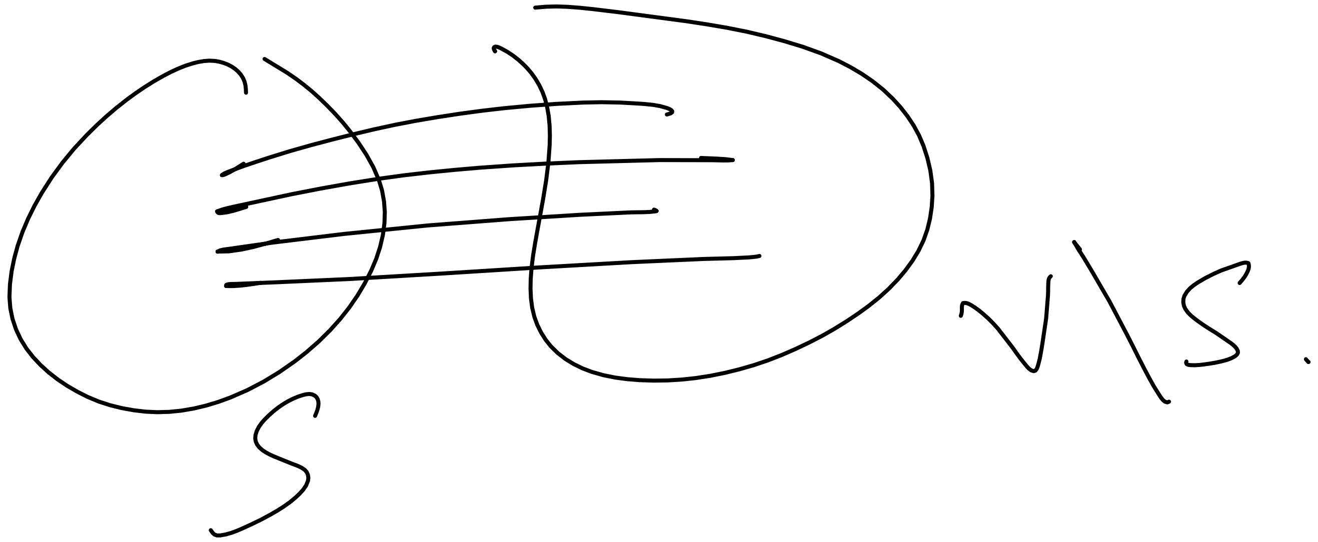
② Clustering



Community detection in Social network.

Def^n (MinCut):

$S \subseteq V$.



$$\underset{S \subseteq V}{\arg\min} \left[\left| \underbrace{E \cap (S \times (V \setminus S))}_{= \mathrm{Cut}(S)} \right| \right]$$

Set of edges that cross cut S

How many cuts are possible?

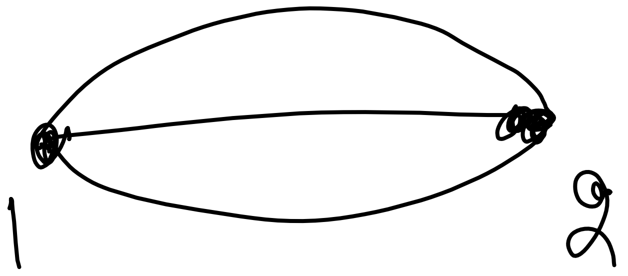
$$\frac{2^n - 2}{2} = \underbrace{2^{\theta(n)}}$$

not going to scale for large n?

Parallel Edges :

no self loops

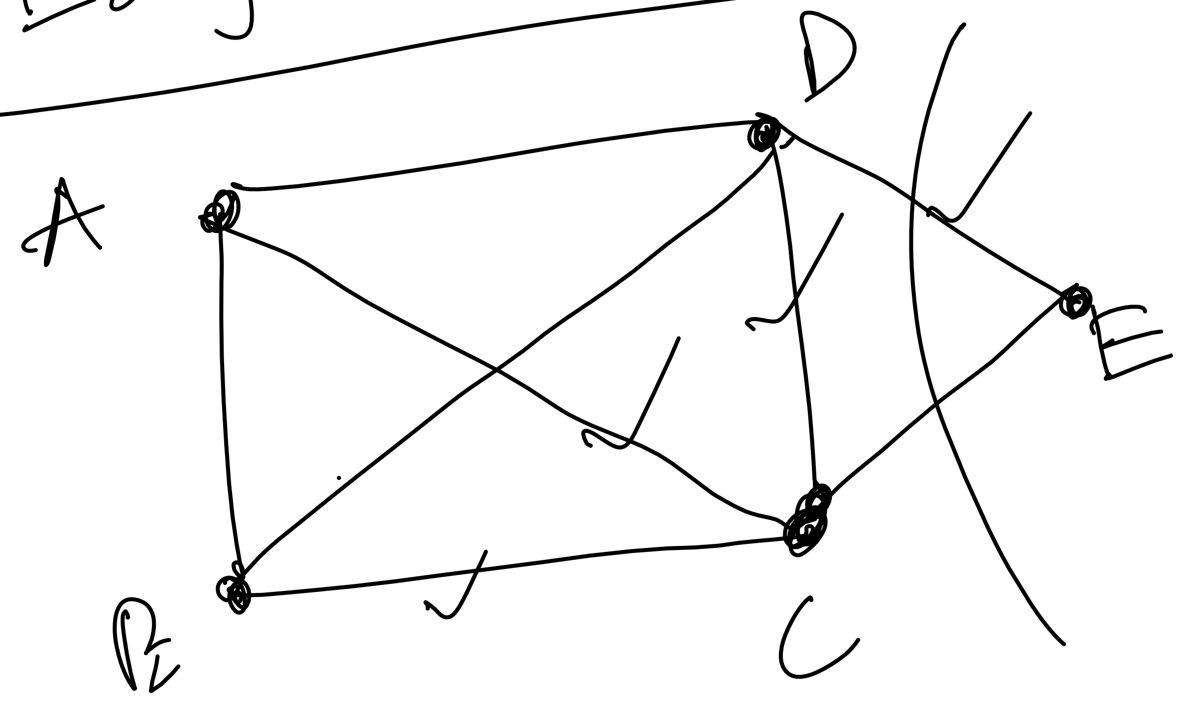
diagonal entries 0.



$$
\begin{array}{c|cc}
 & 1 & 2 \\
\hline
1 & 0 & 3 \\
2 & 3 & 0
\end{array}
\qquad
\begin{bmatrix} 0 & 3 \\ 3 & 0 \end{bmatrix}
$$

$\{0, 1, \dots, \infty\}$.

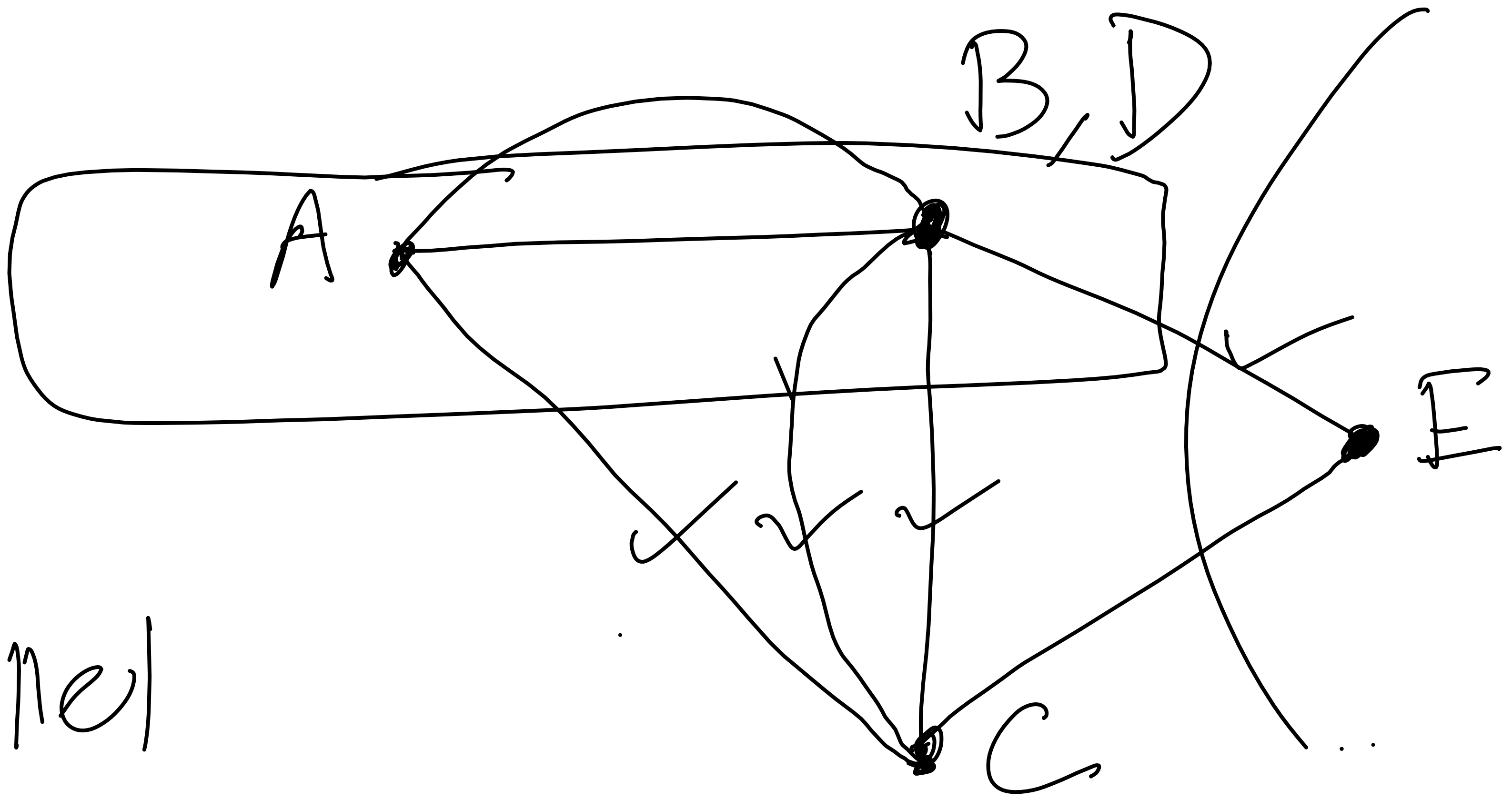
Edge Relaxations :



G

no self loops
but can introduce parallel edges.

(B, D) edge relaxed



Observations: - Any cut in the relaxed graph is also a cut before relaxation in the original graph.

(but not the other way around).

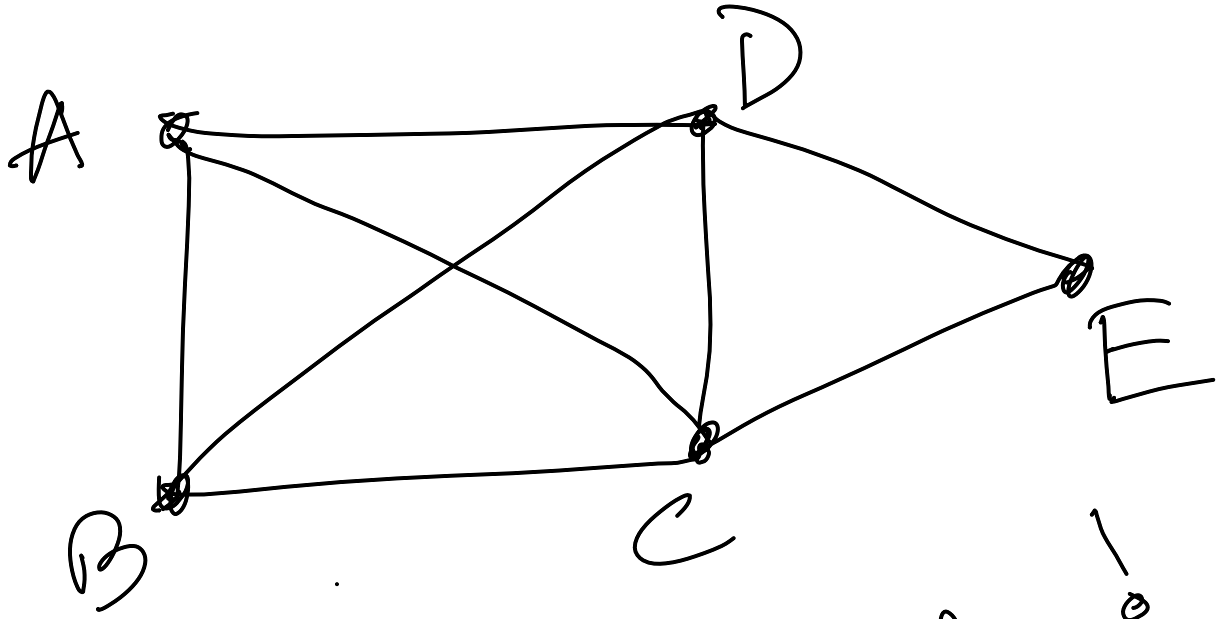
- $\boxed{\text{Relax operation cannot reduce the cut size!}}$

-

Algorithm

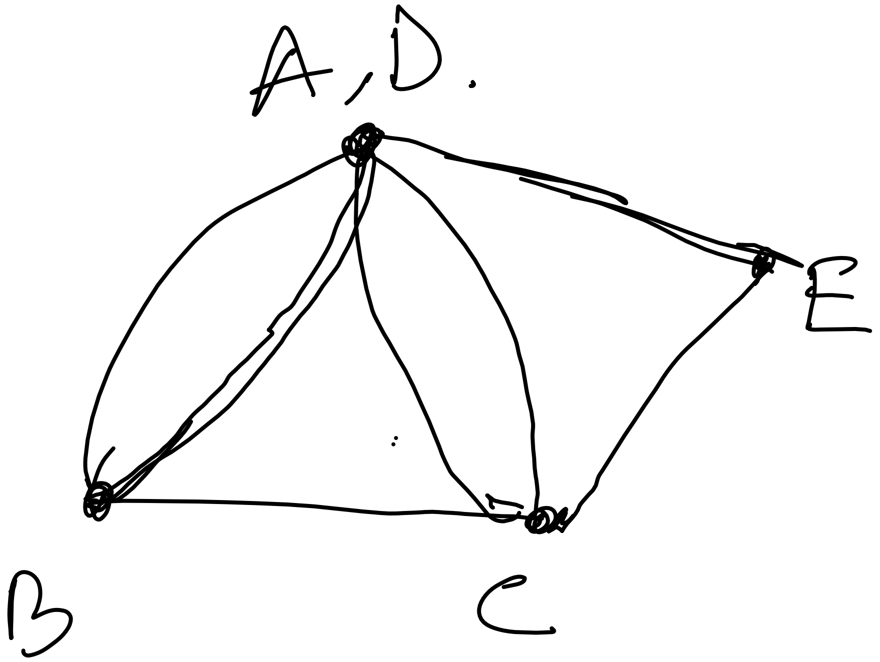
1. Pick one edge (x,y) at random.
2. Relax that edge.
3. Go back to step 1 until only two vertices are left.

Iterate $(n-2)$ time

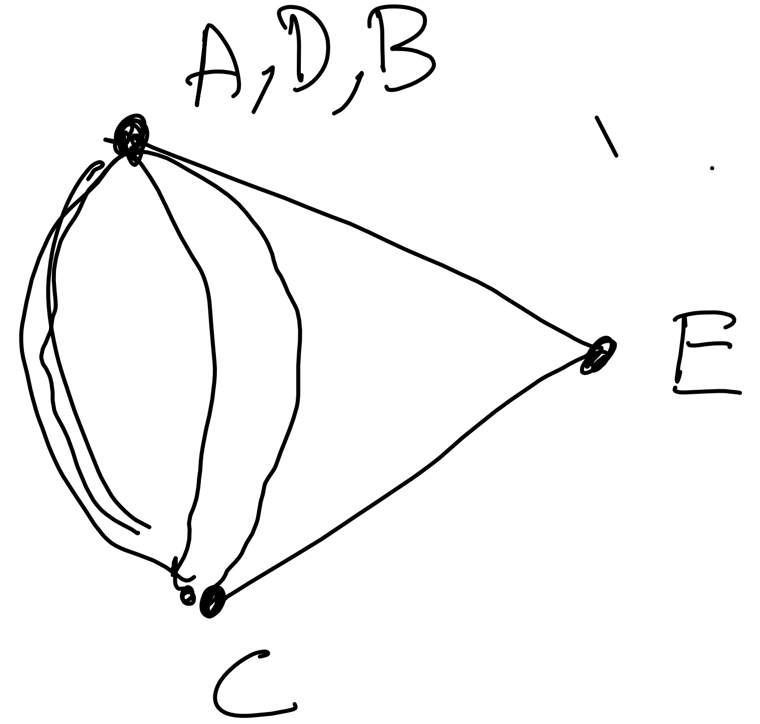


Successful run!

$(A,D) : 1/8$

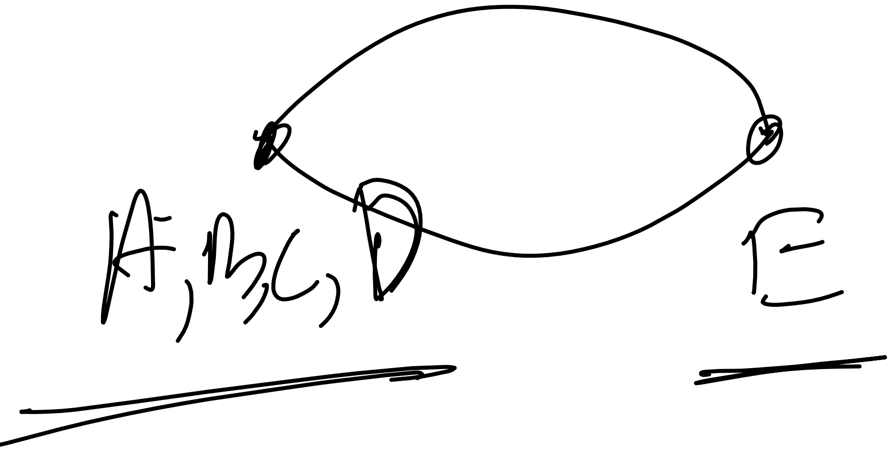


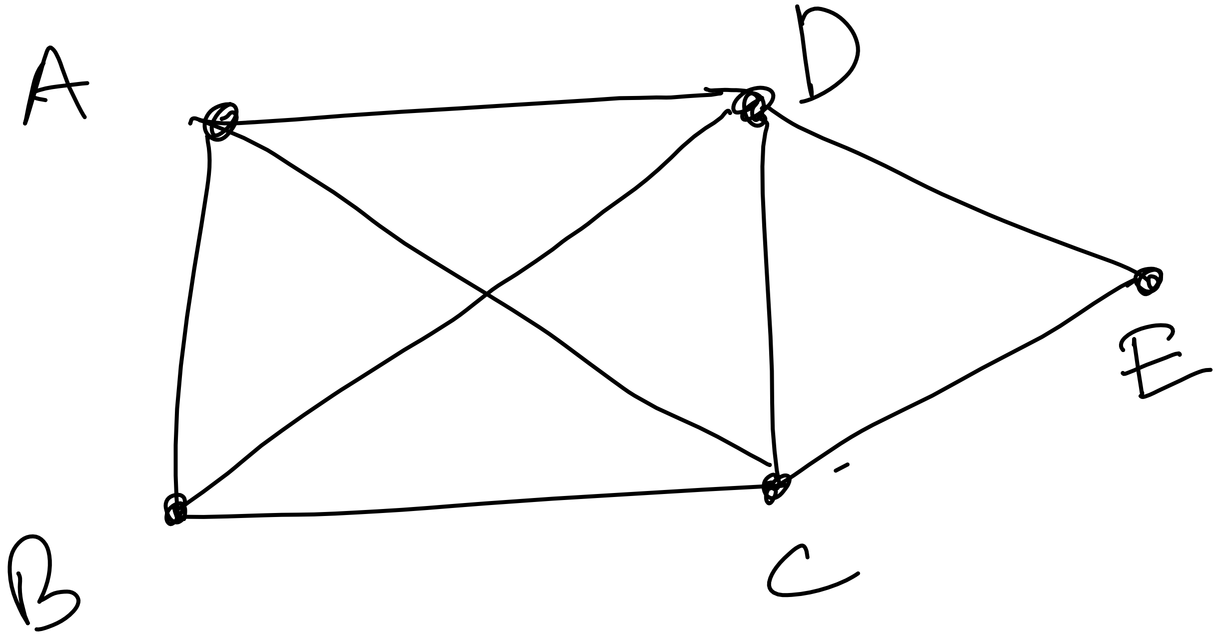
$((A,D), B) : \frac{1}{7}$



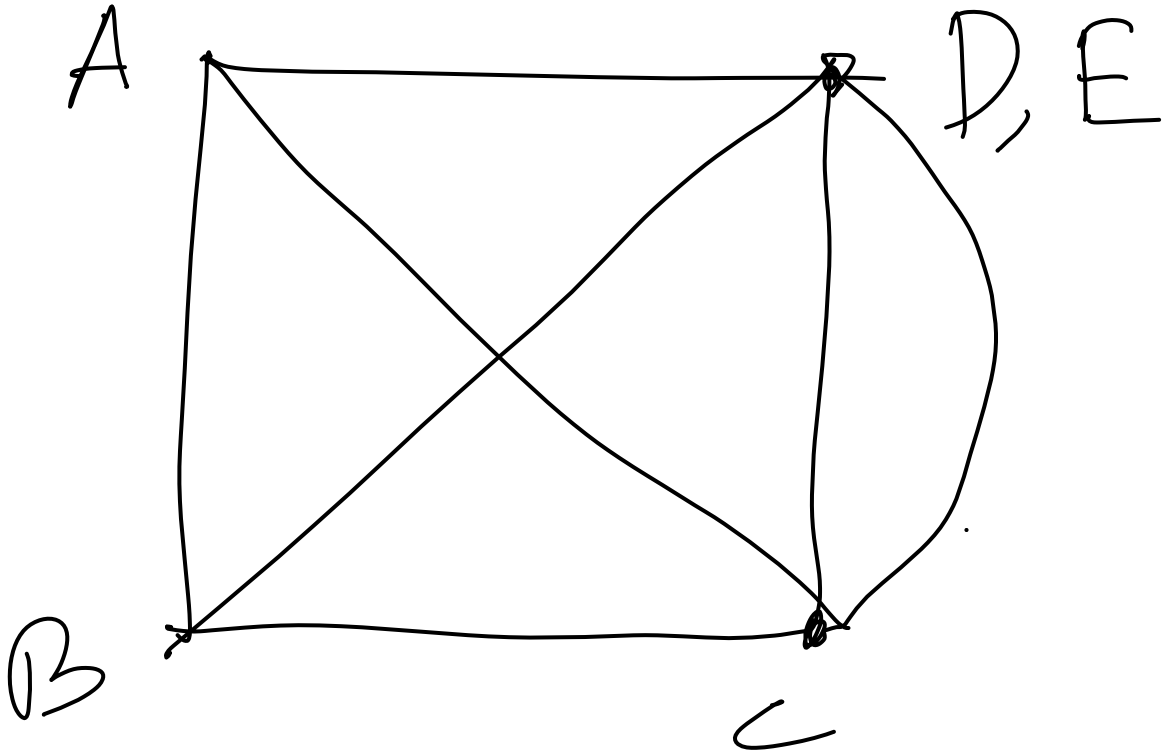
$((A,D,B),C)$

$(1/5)$





$\xrightarrow{(E,D)}$



unsuccessful !

3 or more !

Claim: The cut returned by the algorithm is the ~~in~~ min cut with at least some p probability where $p > 0$.

Proof: Consider any min cut set $\underbrace{Cut(S)}$

set of edges that crosses S to $V \setminus S$.

We'll show that no edge in $Cut(S)$ is relaxed by the algo. w.p. $\geq p$.

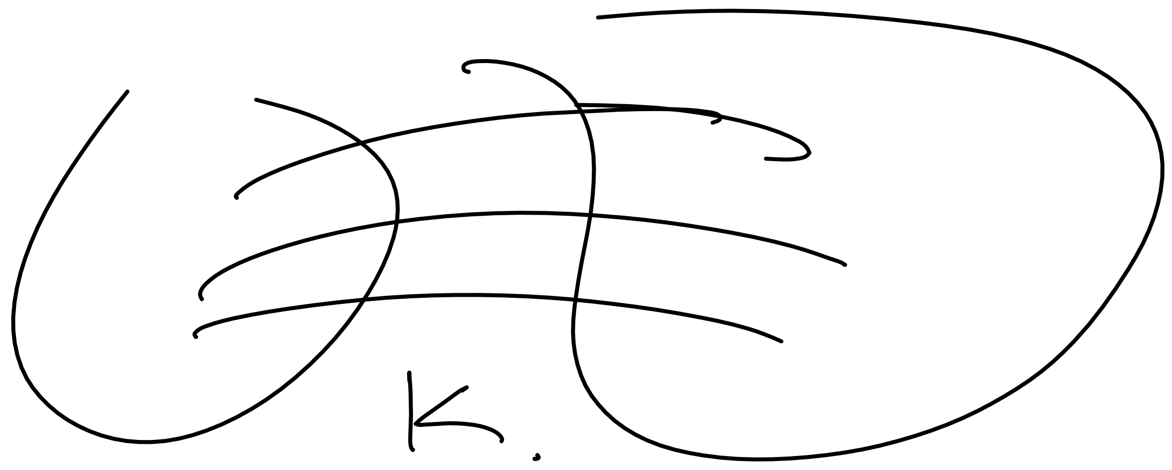
Very first step :

Let K be the size of min cut.

$\Rightarrow$ degree $\geq K$ for each vertex.

* How many edge are there in total in the graph?

$$\geq \frac{nK}{2}.$$



S $\quad$ $V \setminus S$.

min cut ~~(S)~~ S.

Cut (S) is

Prob[any edge in relaxed]

$$\leq \frac{K}{\left(\frac{nK}{2}\right)} = \frac{2}{n}.$$

Second step:

relaxed graph: $(n-1)$ vertices

by prev. obs. relaxed graph also has min. cutsize $\geq K$

Prob. in second step you didn't relax any edge from Cut(S).

$$\leq \frac{K}{\frac{(n-1)K}{2}} = \frac{2}{(n-1)}$$

This holds for any i-th step. $(n-i+1)$ vertices: $\frac{2}{(n-i+1)}$

$E_i \leftarrow$ the event that no edges in $\underline{\underline{Cut(S)}}$ is relaxed at step i

$$E_1 \geq \left(1 - \frac{2}{n}\right)$$

$$F_i \leftarrow \left(E_1 \wedge E_2 \wedge \ldots \wedge E_i\right)$$

the event that no edges in $Cut(S)$ is relaxed up to step i

$$\Pr[F_{n-2}]$$

$$= \Pr[E_{n-2} \wedge F_{n-3}].$$

$$= \Pr[E_{n-2} \mid F_{n-3}] \cdot \Pr[F_{n-3}]$$

$$= \Pr[E_{n-2} \mid F_{n-3}] \cdot \Pr[E_{n-3} \mid F_{n-4}] \cdots$$

$$\cdots \quad \Pr[F_1].$$

$$\Pr[F_1] = \Pr[E_1] \geq \left(1 - \frac{2}{n}\right).$$

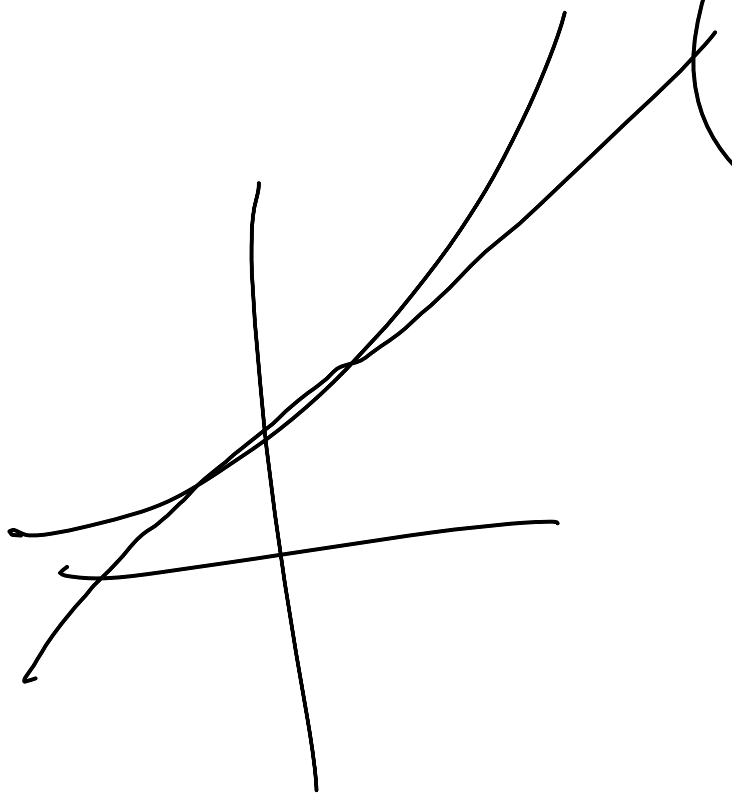
$$\Pr[E_j \mid F_{j-1}] \geq \left(1 - \frac{2}{n-j+1}\right)$$

$$= \left(1 - \frac{2}{n}\right) \cdots \left(1 - \frac{2}{n-j+1}\right) \cdot \underbrace{\left(1 - \frac{2}{3}\right)}_{1/3}.$$

$$= \frac{2}{n(n-1)} \leftarrow$$ cut(S) is still preserved at end of algo. w.p. at least this.

We'll improve the success prob. by repeating this ~~ex~~ ~~rando~~ algorithm t times and take the minimum.

$$\left(1-\frac{2}{n(n-1)}\right)^{t} \le \underline{\underline{\text{small}}}$$



$$\le e^{-\frac{2}{n(n-1)}\cdot t} \le \delta$$

99.9%

$.001$

$\frac{1}{1000}$

$e^x \ge 1+x$

$$t \ge \ln\frac{1}{\delta}\cdot\frac{n(n-1)}{2}$$

$$30 \frac{n(n-1)}{2} = \underline{15n^2}$$

$$t = \underline{O(n^2)}$$