18.453 Lecture 1

Lecture Plan:
- Intros
- Logistics
- ABOUT THE TOPIC
- Breakout rooms to work on examples

Join course on explain.mit.edu!

INTROS:

ABOUT ME:
- COLE FRANKS
  - PLS call me COLE
- Postdoc in applied math
- study theoretical computer science

ABOUT YOU: Pls say your

- name
- Major (in english not numbers)
- year
- Draw yourself in explain.mit.edu main room

Logistics:

- lectures, recorded but attend. encourage.
- one OH W 11-12:30, another TBA.
- Bi-weekly pset. 40%
  1 quiz. 25 %
  1 final 35. %
- Pset in groups: non-mandatory
  write-ups must be done individually.

WHAT IS COMBINATORIAL OPTIMIZATION?

$$\text{find } \max_{x \in X} f(x)$$

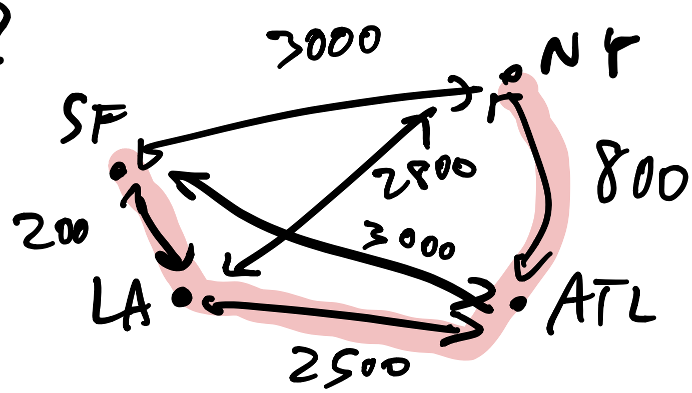
Some set

Some function

- unlike calculus, X usually finite e.g. $\{0,1\}^n$ or discrete e.g. $\mathbb{Z}$.
- However, even when X finite, too hard to check all elts: $|\{0,1\}^n| = 2^n$ ....

Famous example: Travelling Salesman problem (TSP)

— Given pairwise distances between n cities, what's shortest route to visit them all?



\# possible trips : $n! \gg 2^n$

Thus, we need better Techniques than

simply trying all possibilities.

- Frequently this is just impossible. e.g. TSP is "NP hard".
- We still get lucky for many combinatorial structures!
  - Matchings
  - flows / cuts
  - TREES & MATROIDS

- SUBMODULARITY

- <u>Main tool</u>: linear programming (LP)

e.g.

$$\boxed{\begin{aligned} &\max \quad 2x + 3y \\ &\text{subject to} \quad x + y \le 2 \\ &\qquad\qquad\quad x - y \le 4 \\ &\qquad\qquad\quad x \ge 0 \\ &\qquad\qquad\quad y \ge 0. \end{aligned}}$$

- Even when we aren't lucky, LP and other tools can help approximate (e.g. TSP).

Example: Matchings

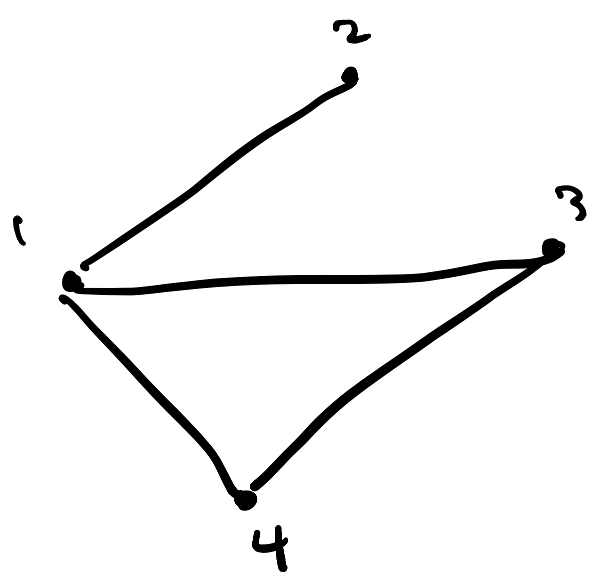
- Graph: $G = (V, E)$

$$\underset{\text{vertices}}{V} \quad \underset{\text{edges}}{E}$$

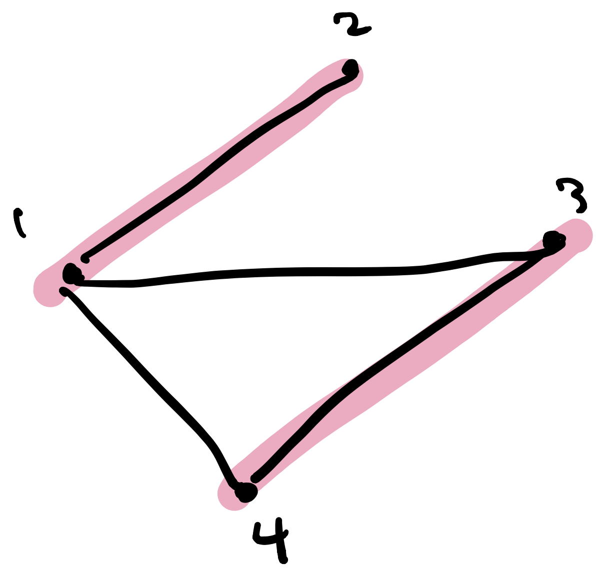
e.g. $V = \{1, 2, 3, 4\}$

$E = \{\{1,2\}, \{3,4\}, \{1,3\}, \{1,4\}\}$

Depicted

$G =$ 

- <u>matching</u>: $M \subseteq E$ disjoint set of edges.

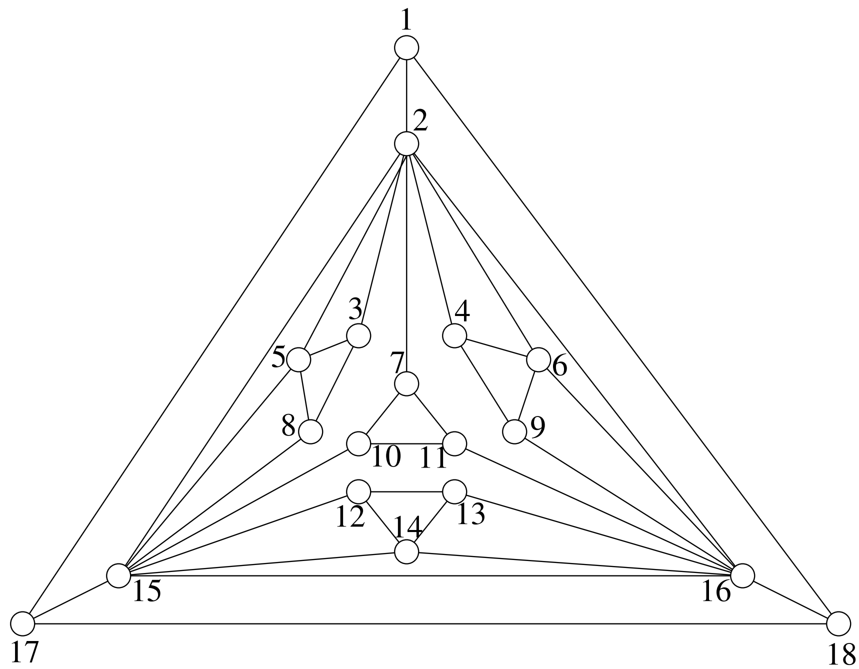


- <u>perfect matching</u>: M includes all vertices (G above has <u>perfect</u> matching).

ACTIVITY 1:

Matching illustration

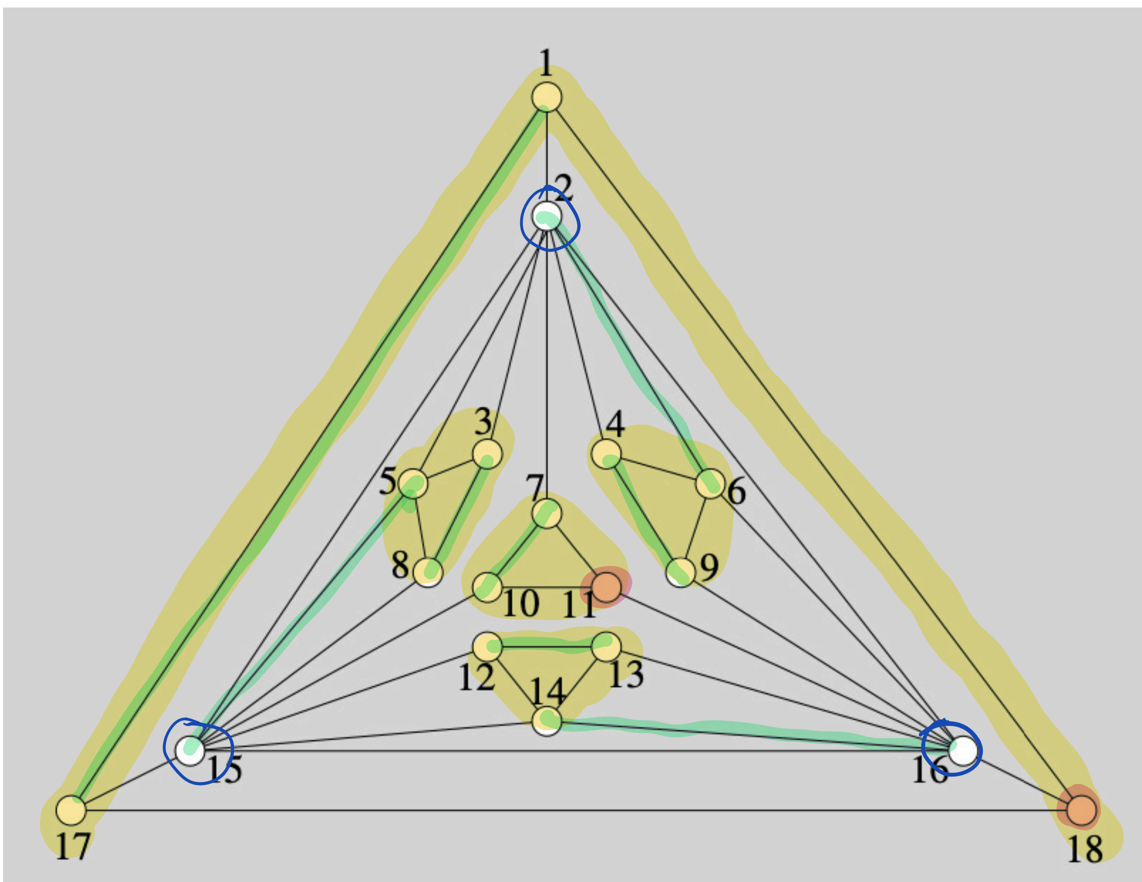
A matching M in a graph $G = (V, E)$ is a set of edges with no endpoints in common. In the graph below, find a matching of maximum size.



How can you convince someone that the matching you found is indeed of maximum cardinality?

Do this w/ your breakout room in explain.mit.edu

Matching w/ n-2 vertices

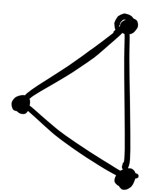


Key Theme: Duality

loosely: The SIMPLE obstructions are the ONLY obstructions

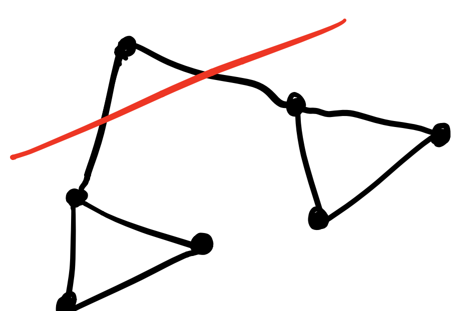
- What OBSTRUCTS matchings?

  - parity



  - parity + cuts

at least 1 vertex unmatched.



This is the only kind of obstruction!

Tutte's theorem: If $U \subseteq V$, let

$$o(U) = \{\#\text{ odd connected components if } U \text{ is removed}\}$$

Then

$$\max \left| \overset{\text{vertices}}{\text{matching in } G} \right| = \min_{U \subseteq V} |V| + |U| - o(U).$$

Eventually we'll show how duality leads to efficient algs for matching!

Activity No. 2:

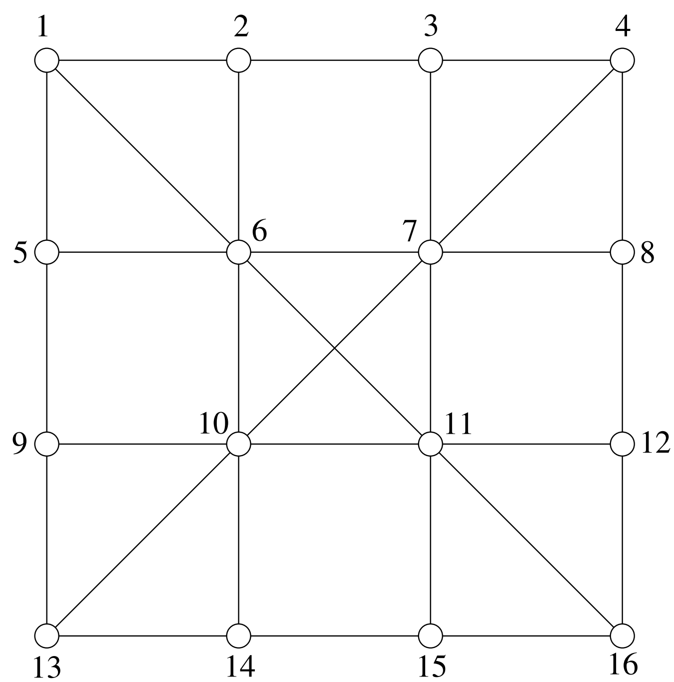
Massachusetts Institute of Technology 18.453: Combinatorial Optimization
Instructor: Cole Franks **Notes:** Michel Goemans and Zeb Brady February 15, 2021

Spanning Tree Game

A spanning tree T in a graph $G = (V, E)$ is a set of edges without any cycles that connect all vertices together. The spanning tree game is a 2-player game. Each player in turn selects an edge. Player 1 starts by deleting an edge, and then player 2 fixes an edge (which has not been deleted yet); an edge fixed cannot be deleted later on by the other player. Player 2 wins if he succeeds in constructing a spanning tree of the graph; otherwise, player 1 wins.

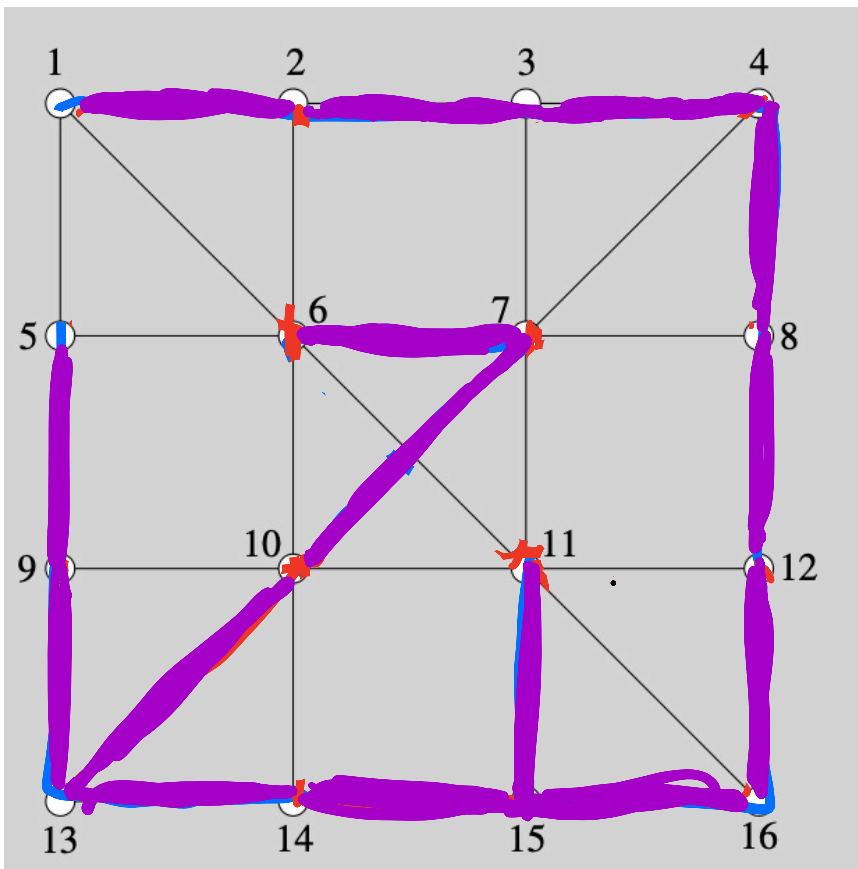
* Which graphs have a winning strategy for player 1? Which graphs have a winning strategy for player 2?

For this graph with 16 vertices and 30 edges, which player has a winning strategy?

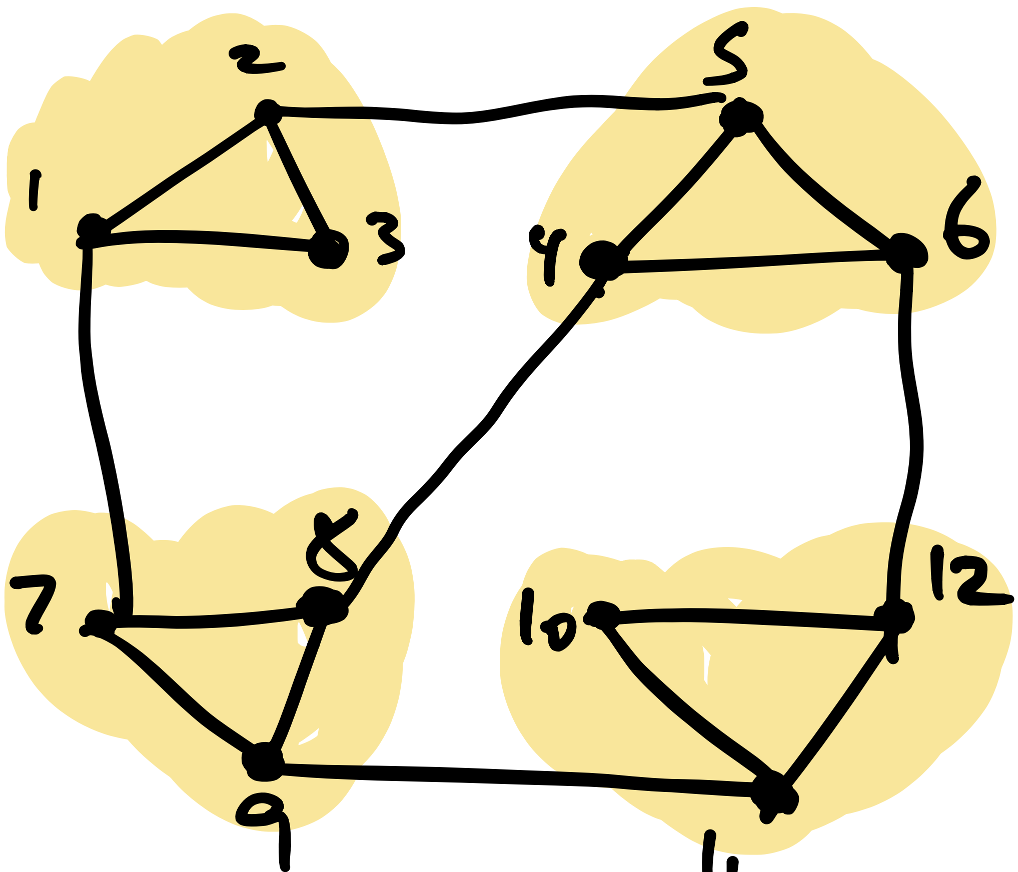


- Play on this graph w/ your group.
- Find examples where P1 wins & where P2 wins.

• Try to answer *.



another example:



CASE 1: $\exists$ 2 disjoint spanning trees $A, B \in G$.

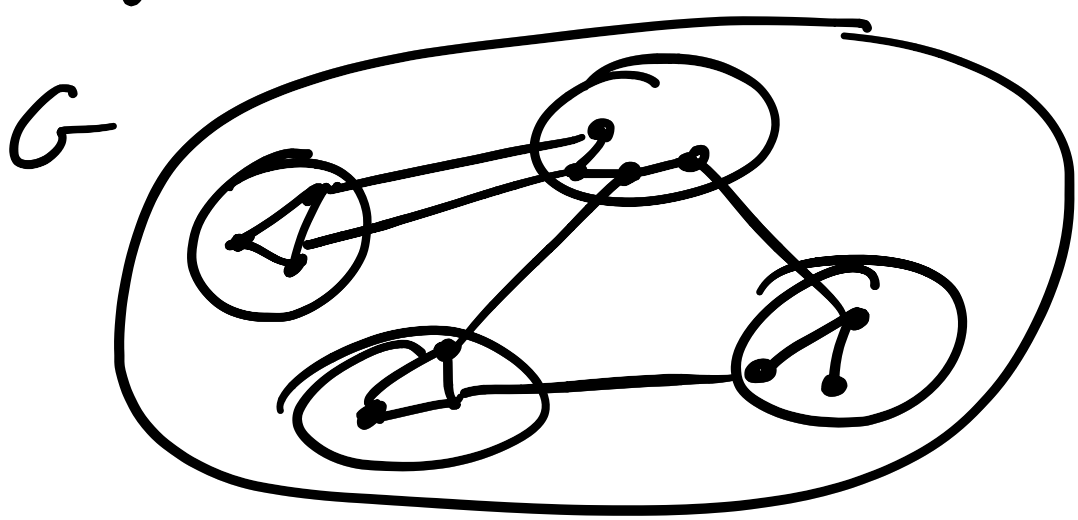
Claim: P1 wins!

- when P1 cuts from A, P2 adds edge from B to A so A is still spanning tree. "Exchange property"
- A, B will be disjoint except fixed edges.
- in the end, A = B is spanning tree remaining.

CASE 2: no 2 disjoint spanning trees.

claim: P2 wins!

Duality: simple obstruction for 2 disjt. spanning trees.



4 parts, only

$2 \cdot (4-1) - 1 = 5$

edges between

them. But a

spanning tree would

have $\geq p-1$ edges

between p parts –

& 2 spanning trees

would have $2(p-1)$

edges!

Thus, partition into p parts w/ $< 2(p-1)$ edges between the parts is an obstruction to 2 disjoint spanning trees.

Thm (Lehman): This is the ONLY obstruction - duality!

Back to Case 2:

Show P1 wins:

- no 2 disjt spanning trees,
  $\Rightarrow \exists$ partition into p parts
  w/ $< 2(p-1)$ edges between
  parts.

- P2 can delete $\geq \frac{1}{2}$ of

these - not enough left to
connect up the parts. $\square$

Algorithmically? How to
find the trees/partition?

Spanning trees example of matroid;

(set system w/ "exchange property";

generalizes set of bases of vector space)

2 disjt. spanning trees example of matroid intersection

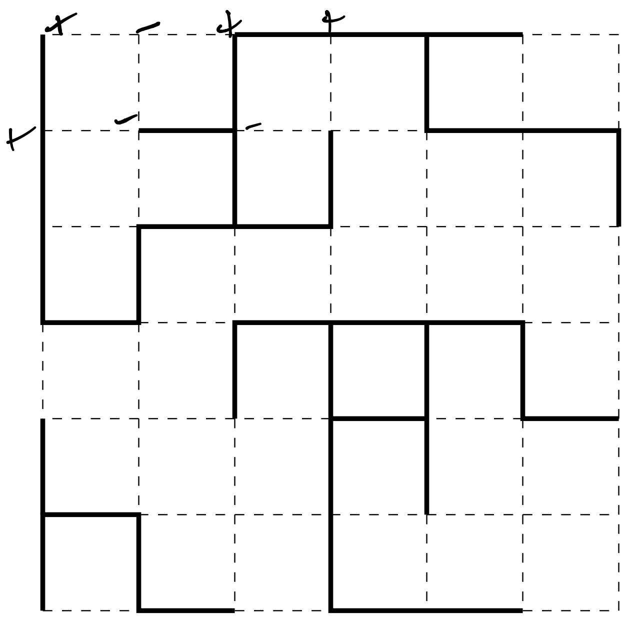
which we will solve later in the course

Activity 3:

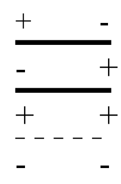
Spin Glass

Consider the 7×7 grid drawn below, where each edge has been made thick (solid) or thin (dashed). Given an assigment of signs (+ or -) to the vertices of this grid, a thick edge is *violated* if the endpoints have two different signs. A thin edge is violated if the endpoints have identical signs. The goal is to find an assignment of signs which minimize the total number of violations in the grid.

$\Pr \propto e^{-\#\text{violations}}$



Violations if



As you'll probably realize, although finding a "good" assignment of signs might not be that difficult, providing a proof that your solution is optimal is in fact much more challenging (and a short proof exists for any instance).

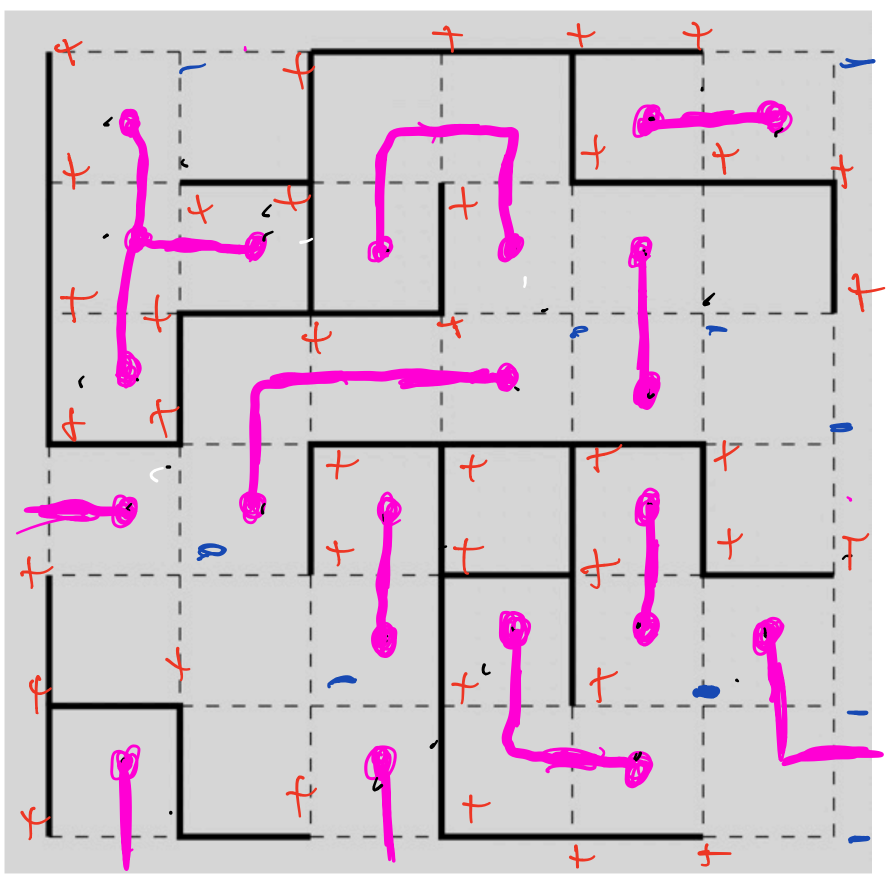
In your groups, try to find the best assignment you can, and try to see what LOWER bounds you can prove.

Turns out: reduces to weighted perfect matching!

Idea: • draw ● on squares w/ odd # thin edges.

(called "frustrated placquettes"...)

• draw | across violated edges.

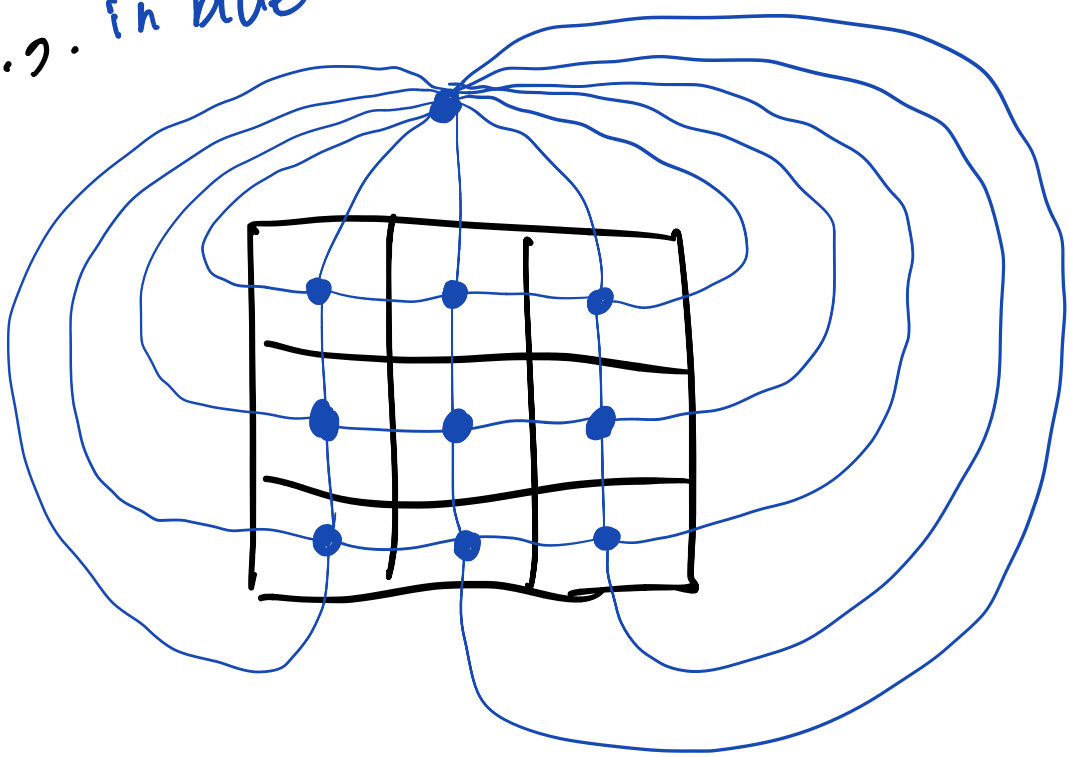


Note: pink edges form graph. G
degree of frustrated placquettes in G is ≥ 1.

Make Dual graph:

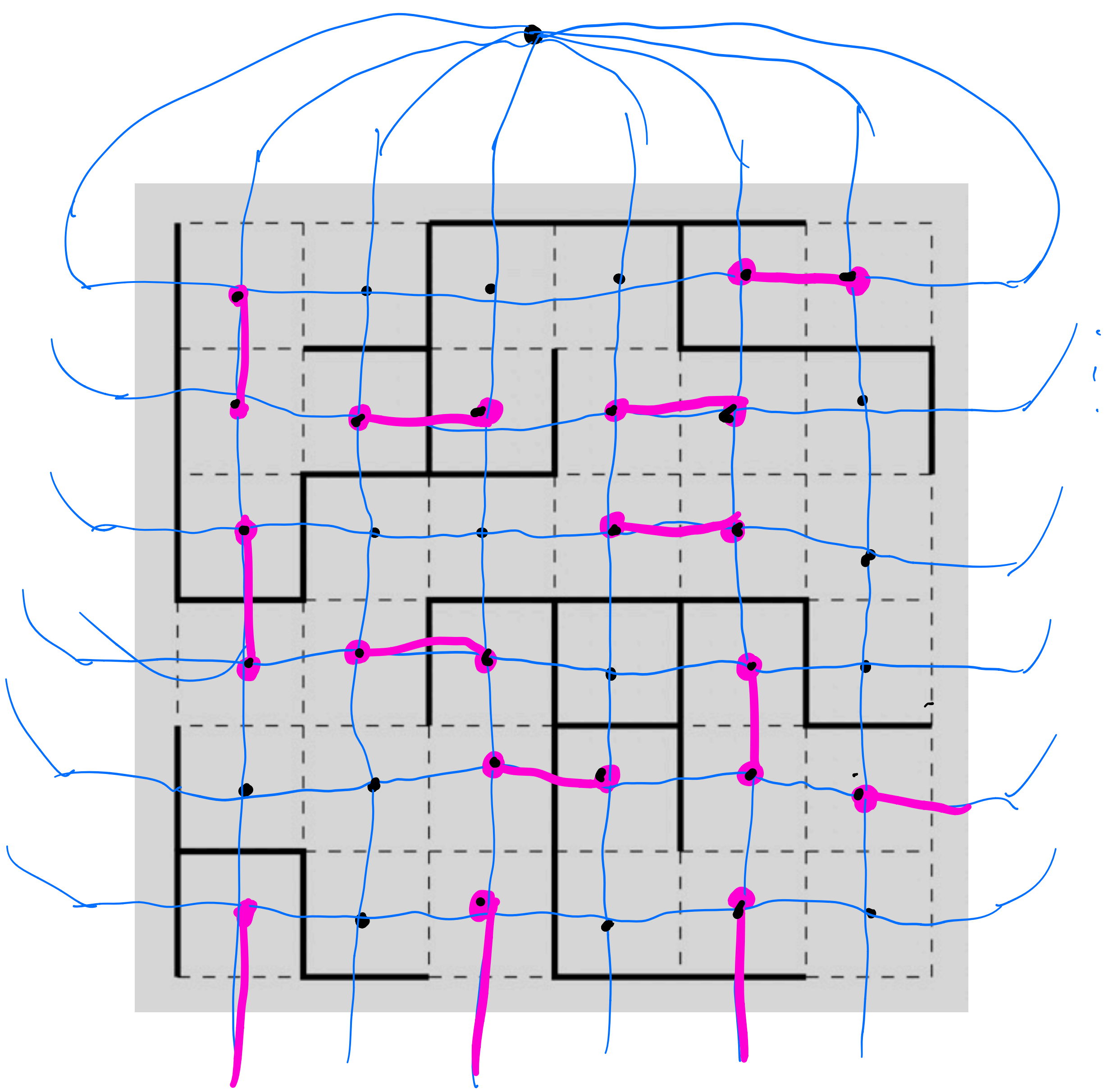
each square is vertex, edges b/w neighboring squares, one vertex for outside.

e.g. in blue



TURNS OUT: min # violations is just min # edges of graph

- w/ odd degree on frustrated placquettes.
- even degree on unfrustrated.

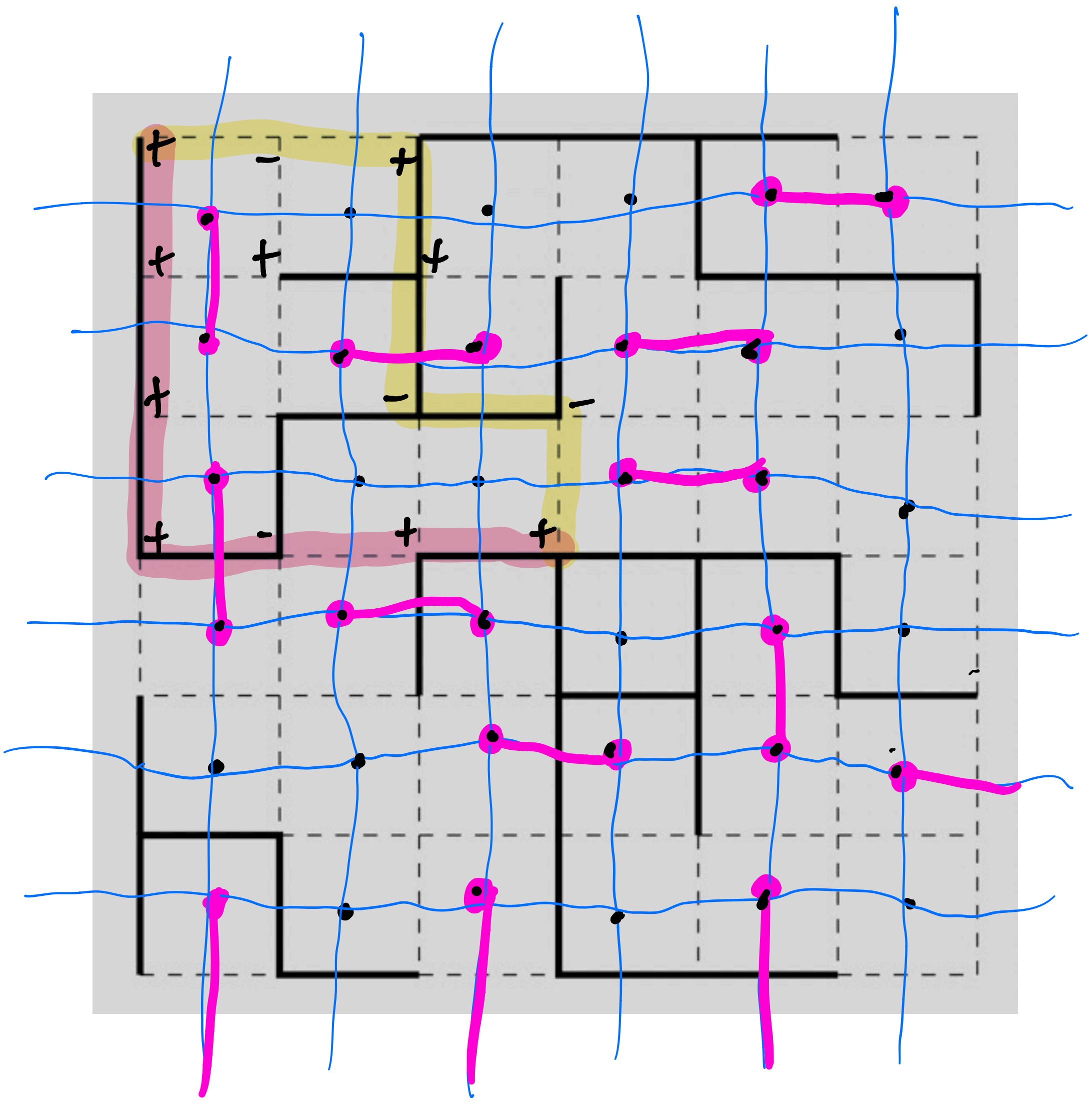


Can assume it's just a union of edge disjoint chains;

Cost is min cost matching in **weighted** graph G with

$V =$ Frusted placquettes
$w(u,v) =$ distance between u,v in <u>dual</u> graph.

How to get signs?



how do we know <u>consistent</u>?

because of parity of degree?