CSE 421
Introduction to Algorithms

Lecture 20: Linear Programming:
A really very extremely big hammer

Note: Wed Nov 22
will introduce a
new topic!

Midterm

Midterm grades will be released at the end of this class.

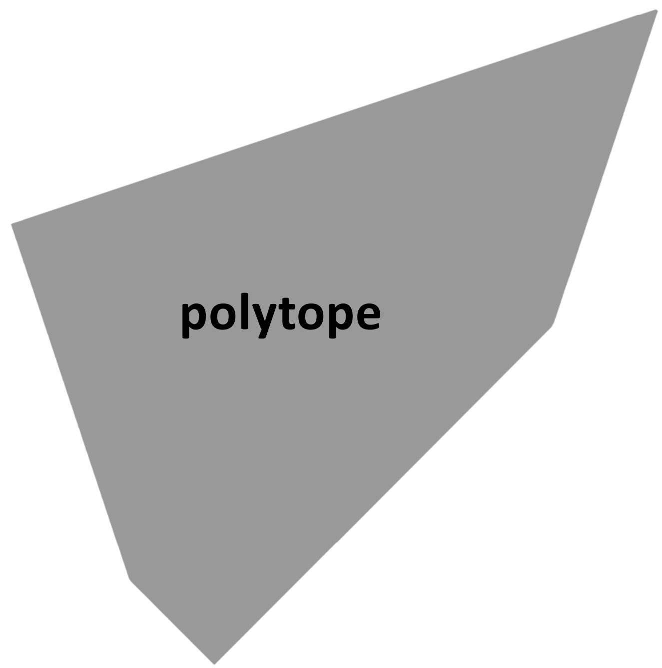
- Breathe!
- These grades don't count for that much.
  - In past 421 I've had a student with a midterm grade in the mid-60's end with a 4.0 in the course

Histogram

90s	**10**	Median	**74**
80s	**24**		
70s	**23**	Average	**67.71**
60s	**13**		
50s	**10**	Standard Deviation	**20.32**
40s	**11**		
<40	**10**		

Given: a polytope

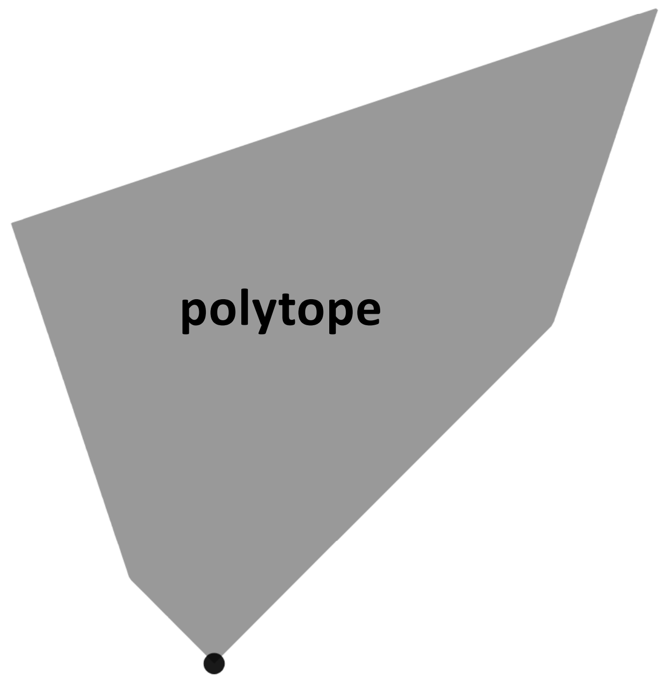
Find: the *lowest* point in the polytope



3

Given: a polytope

Find: the *lowest* point in the polytope



We have fast algorithms for this!

Maximize $z_1 + 2z_3$

subject to:

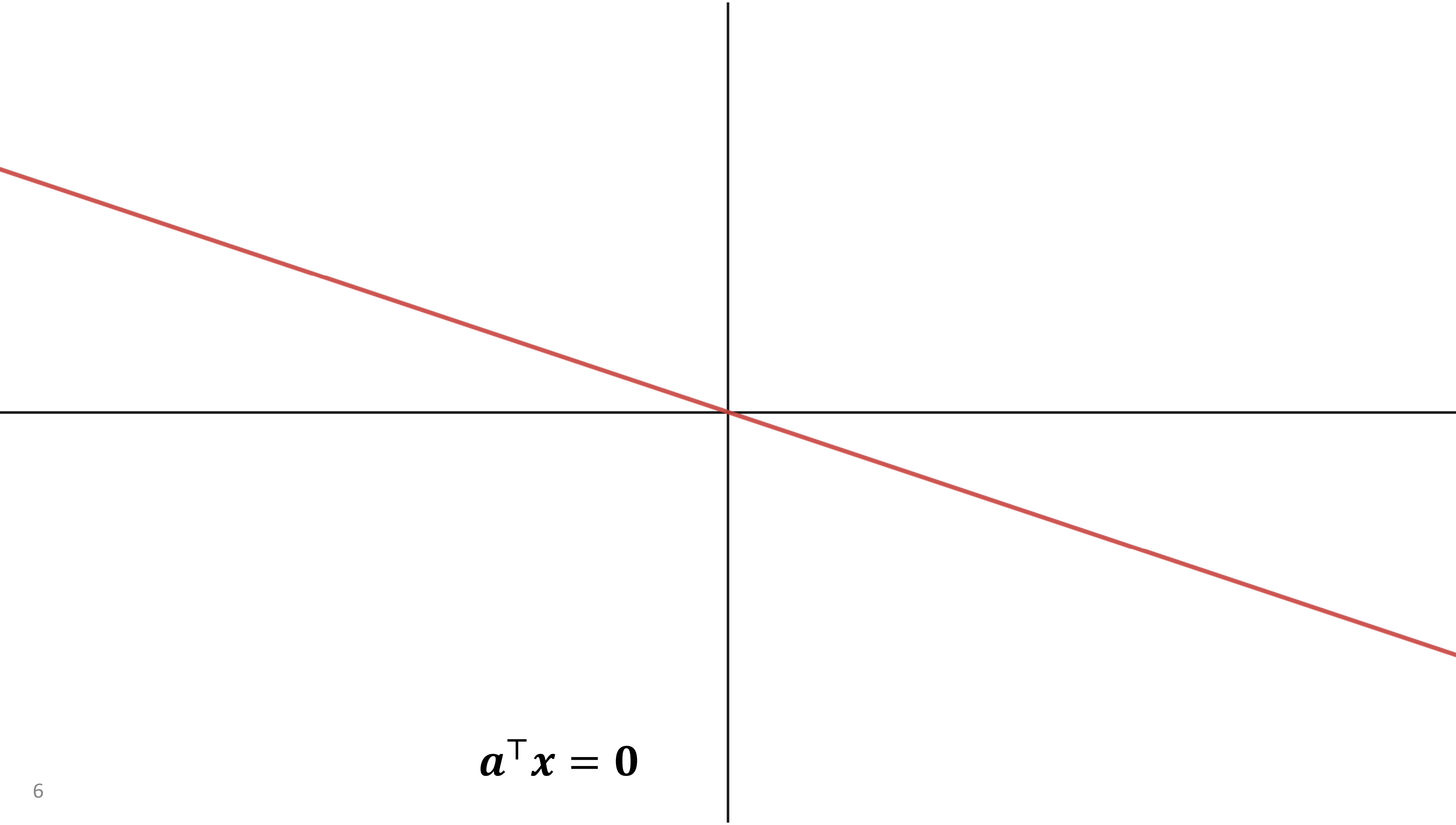
$$2z_1 - z_2 + 3z_3 \le 1$$
$$-z_1 + z_2 - z_3 \le 5$$

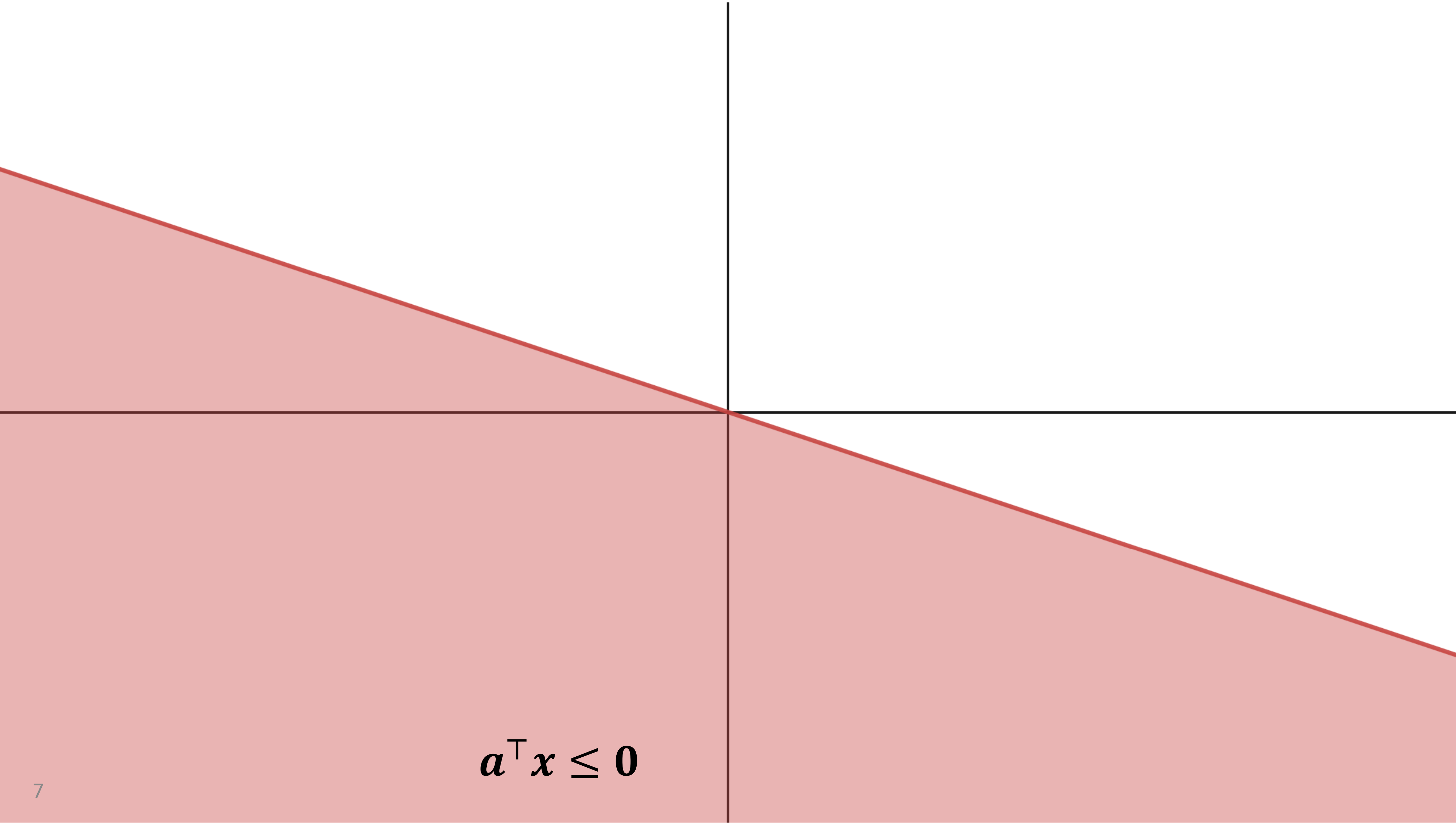
Linear Algebra primer

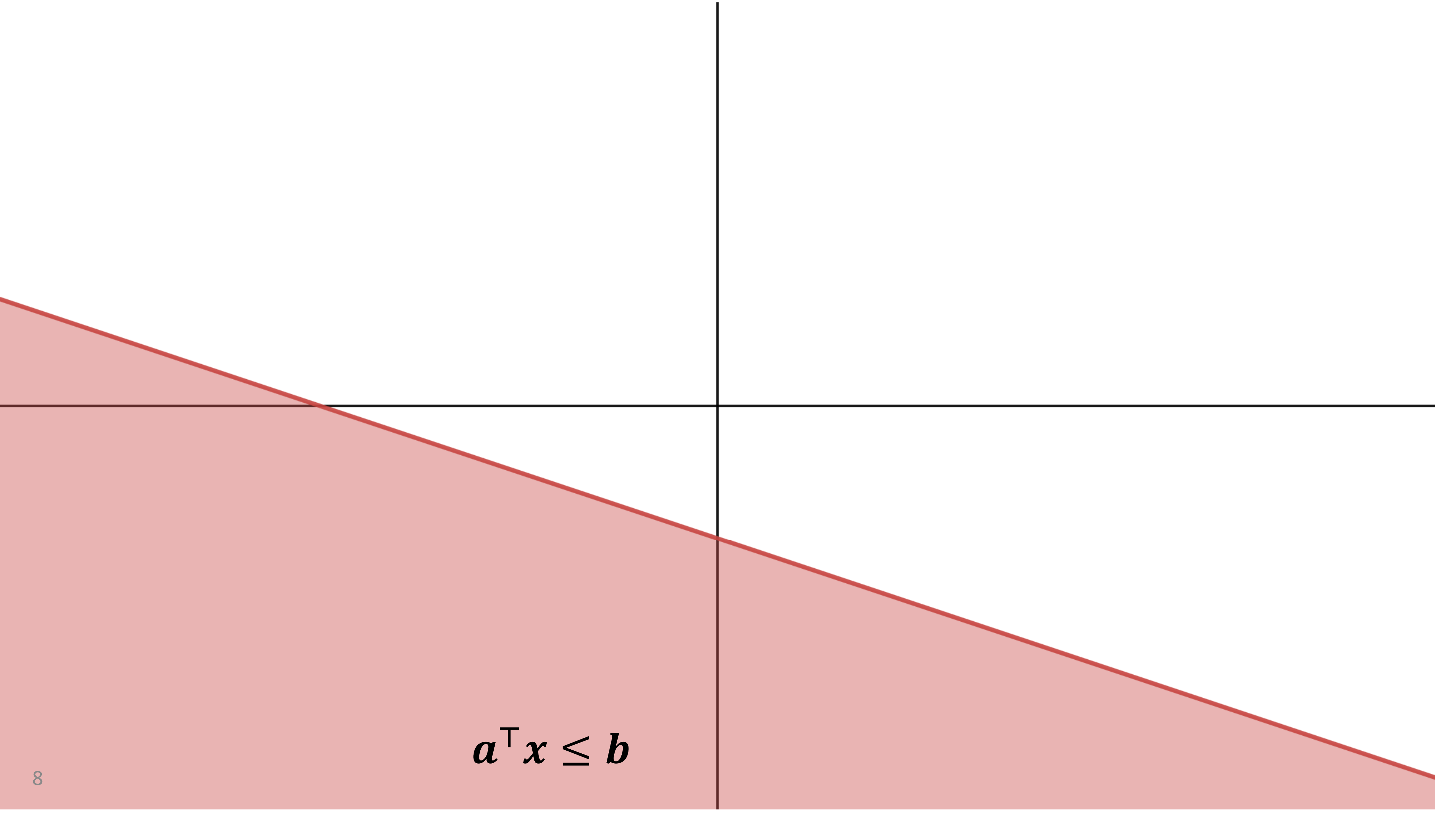
For $\boldsymbol{a}, \boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}}$ we think of $\boldsymbol{a}$ and $\boldsymbol{x}$ as column vectors

$$\boldsymbol{a}^\top \boldsymbol{x} = \boldsymbol{a_1 x_1} + \cdots + \boldsymbol{a_n x_n}$$

The set of $\boldsymbol{x}$ satisfying $\boldsymbol{a}^\top \boldsymbol{x} = \boldsymbol{0}$ is *hyperplane*

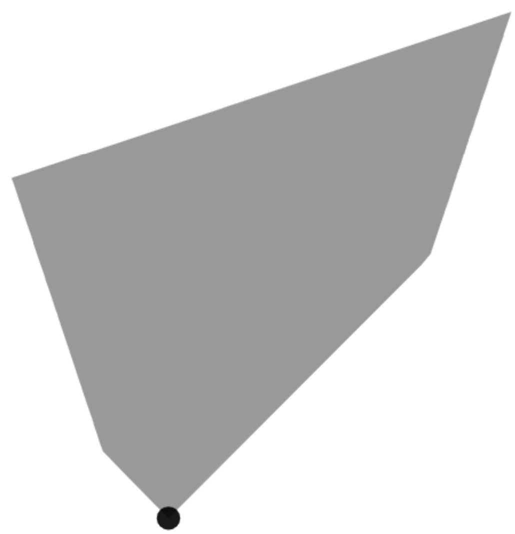

$a^\top x = 0$


$a^\top x \leq 0$


$\boldsymbol{a}^\top \boldsymbol{x} \leq \boldsymbol{b}$

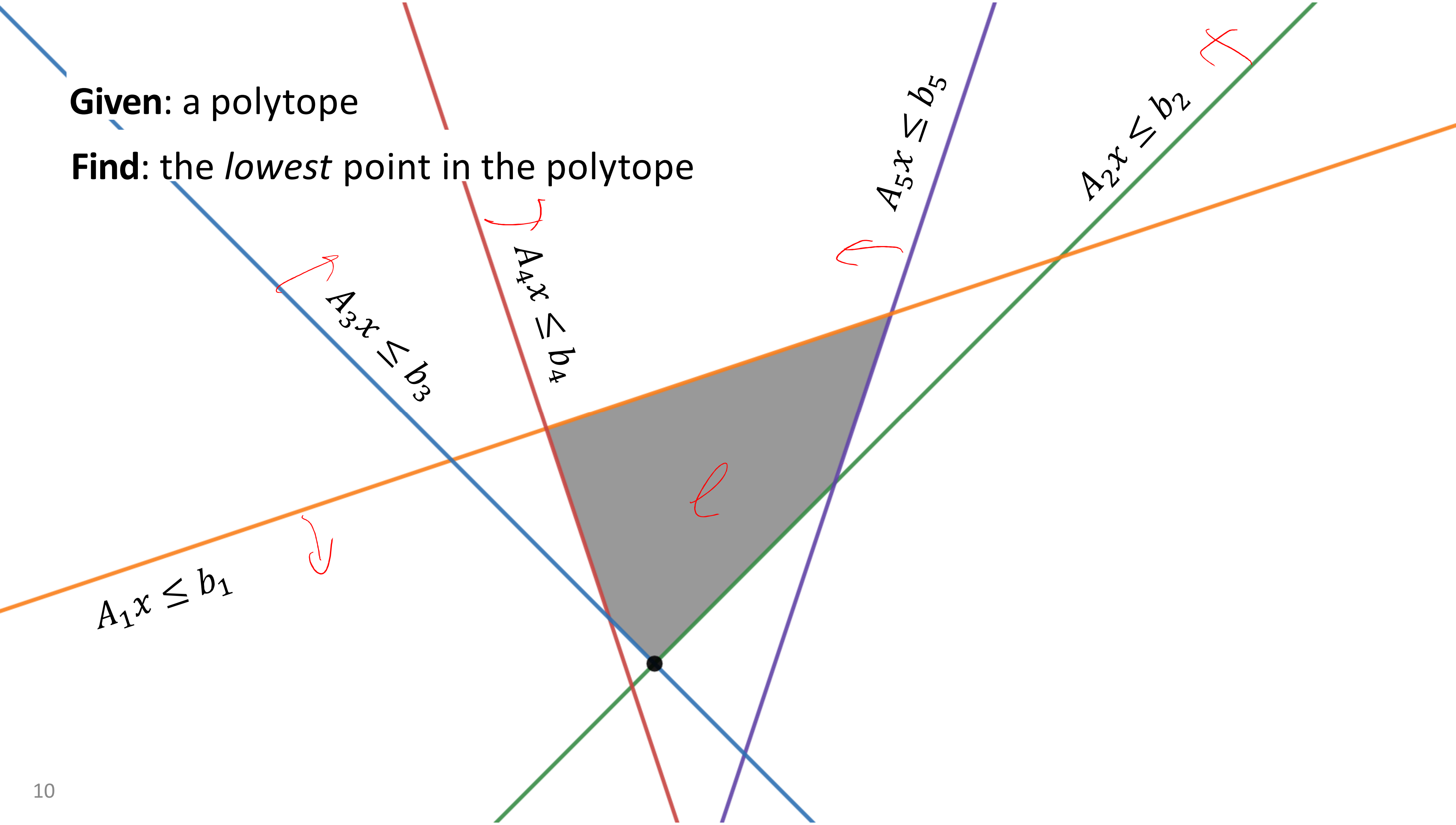
Given: a polytope

Find: the *lowest* point in the polytope



Given: a polytope

Find: the *lowest* point in the polytope



Linear Algebra primer

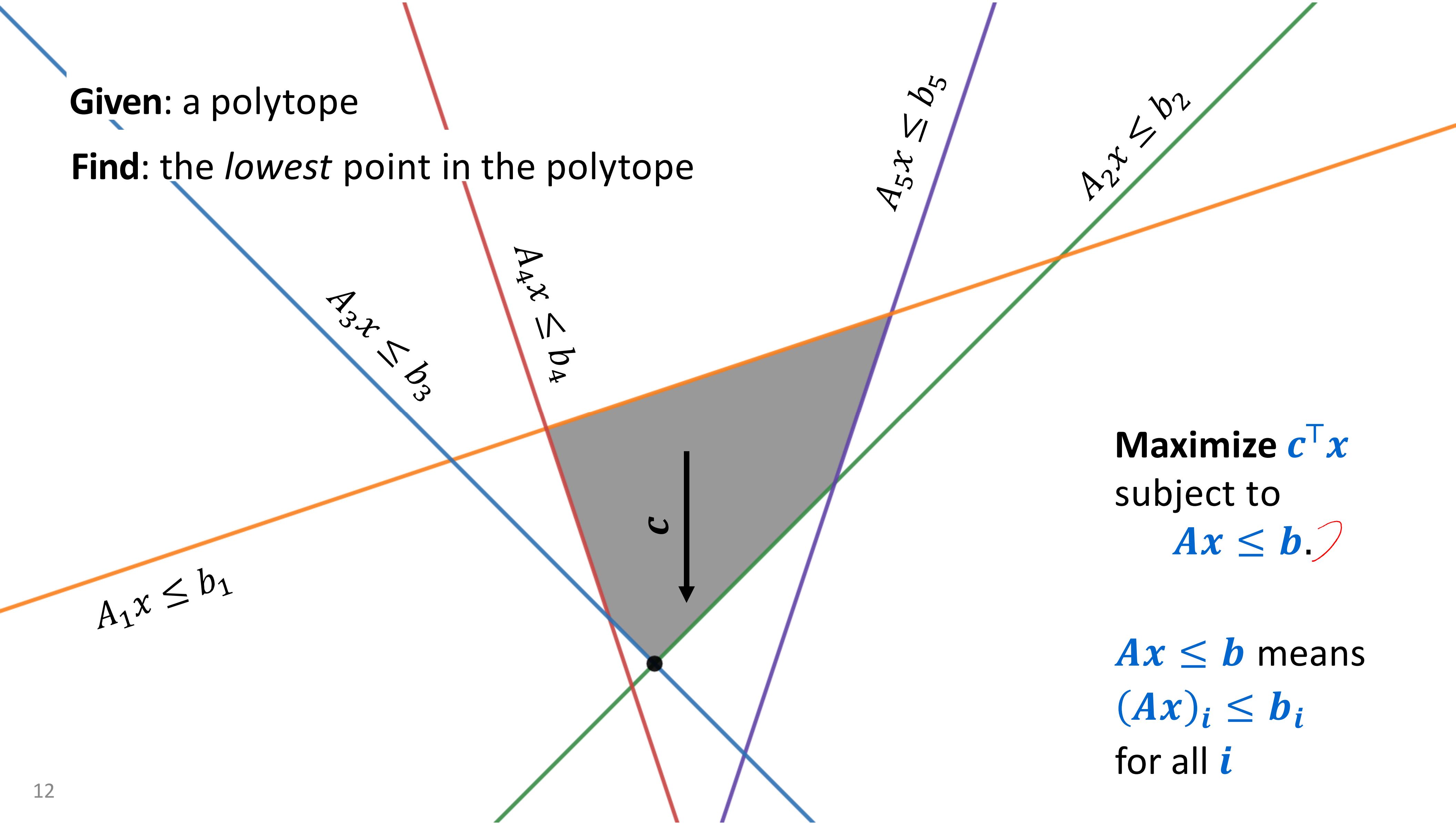
For $\boldsymbol{a}, \boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}}$ we think of $\boldsymbol{a}$ and $\boldsymbol{x}$ as column vectors

$$\boldsymbol{a}^\top \boldsymbol{x} = \boldsymbol{a_1 x_1} + \cdots + \boldsymbol{a_n x_n}$$

Write $\boldsymbol{m} \times \boldsymbol{n}$ matrix $\boldsymbol{A}$, for $\boldsymbol{Ax} = \begin{bmatrix} \boldsymbol{A_1 x} \\ \boldsymbol{A_2 x} \\ \boldsymbol{A_3 x} \\ \cdots \\ \boldsymbol{A_m x} \end{bmatrix}$ where $\boldsymbol{A_1}, \dots, \boldsymbol{A_n}$ are rows of $\boldsymbol{A}$.

Given: a polytope

Find: the *lowest* point in the polytope

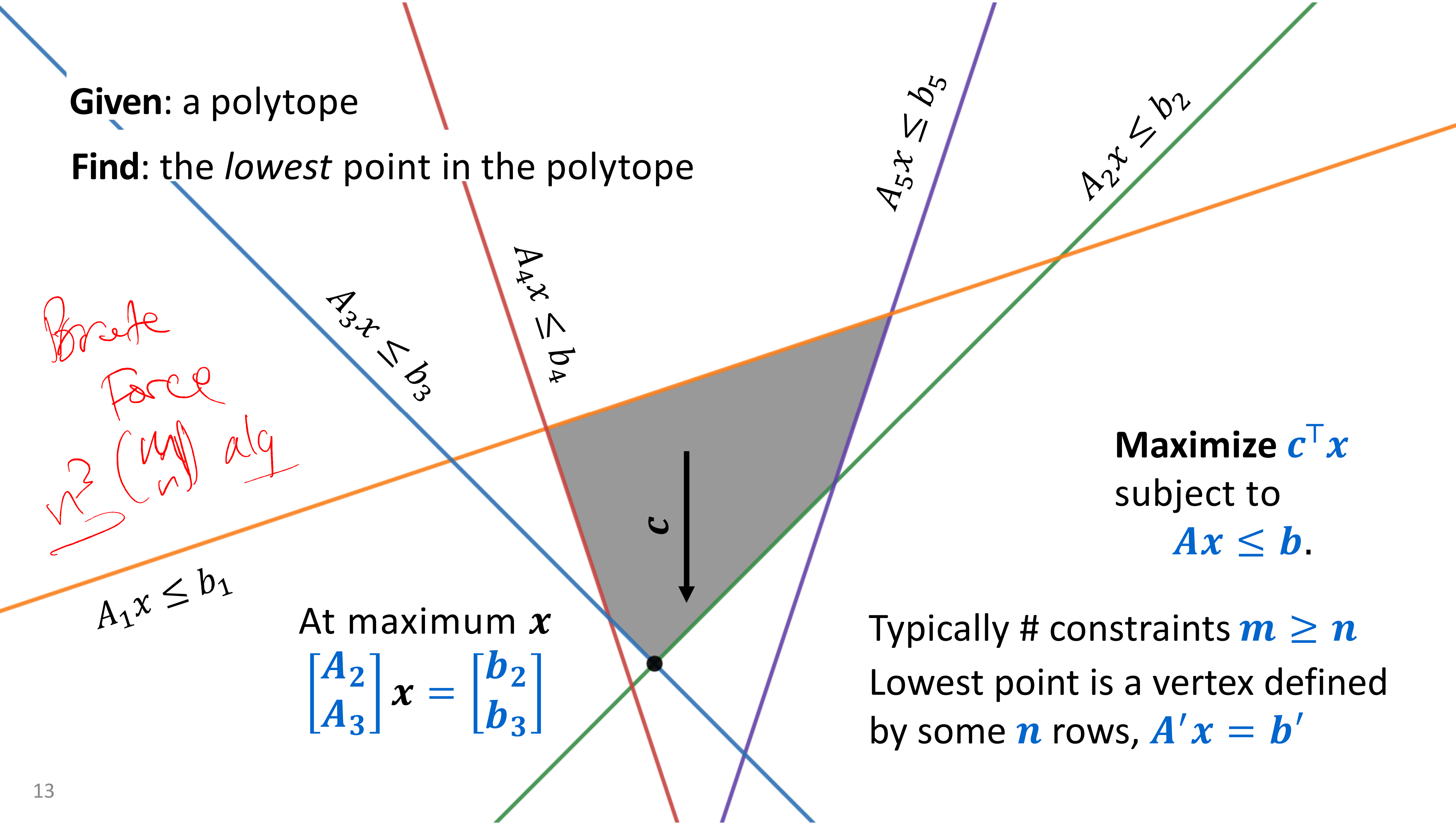


Maximize $\boldsymbol{c}^\top \boldsymbol{x}$
subject to
$\boldsymbol{Ax} \le \boldsymbol{b}$.

$\boldsymbol{Ax} \le \boldsymbol{b}$ means
$(\boldsymbol{Ax})_{\boldsymbol{i}} \le \boldsymbol{b}_{\boldsymbol{i}}$
for all $\boldsymbol{i}$

Given: a polytope

Find: the *lowest* point in the polytope



Brute Force

$n^3 \binom{m}{n}$ alg

At maximum $\boldsymbol{x}$

$$\begin{bmatrix} \boldsymbol{A_2} \\ \boldsymbol{A_3} \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \boldsymbol{b_2} \\ \boldsymbol{b_3} \end{bmatrix}$$

Maximize $\boldsymbol{c}^\top \boldsymbol{x}$
subject to
$\boldsymbol{Ax} \le \boldsymbol{b}$.

Typically # constraints $\boldsymbol{m} \ge \boldsymbol{n}$

Lowest point is a vertex defined by some $\boldsymbol{n}$ rows, $\boldsymbol{A'x} = \boldsymbol{b'}$

Standard Form

$c = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}$

Maximize $z_1 + 2z_3$
subject to
$$2z_1 - z_2 + 3z_3 \le 1$$
$$-z_1 + z_2 - z_3 \le 5$$

$A = \begin{bmatrix} 2 & -1 & 3 \\ -1 & 1 & -1 \end{bmatrix}$

$b = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$

Maximize $c^\top x$
subject to
$$Ax \le b$$
$$x \ge 0$$

replace each z_i by
$x_{i,a} - x_{i,b}$
for $x_{i,a}, x_{i,b} \ge 0$

Maximize $(x_{1,a} - x_{1,b}) + 2(x_{3,a} - x_{3,b})$
subject to
$$2(x_{1,a} - x_{1,b}) - (x_{2,a} - x_{2,b}) + 3(x_{3,a} - x_{3,b}) \le 1$$
$$-(x_{1,a} - x_{1,b}) + (x_{2,a} - x_{2,b}) - (x_{3,a} - x_{3,b}) \le 5$$
$$x \ge 0$$

Max Flow

Given: A Flow Network $G = (V, E)$ with source s, sink t, and $c: E \to \mathbb{R}^{\geq 0}$

Maximize flow out of s

subject to

- respecting capacities
- flow conservation at internal nodes

LP Variables:

x_e for each $e \in E$ representing flow on edge e

Maximize $\displaystyle\sum_{e \text{ out of } s} x_e$

subject to

$0 \leq x_e \leq c(e)$ for every $e \in E$

$$\sum_{e \text{ out of } v} x_e = \sum_{e \text{ into } v} x_e$$

for every node $v \in V - \{s, t\}$

Max Flow

Maximize $\sum_{e \text{ out of } s} x_e$

subject to

$\mathbf{0} \le x_e \le c(e)$ for every $e \in E$

$$\sum_{e \text{ out of } v} x_e = \sum_{e \text{ into } v} x_e$$

for every node $v \in V - \{s, t\}$

Replace equality constraints by a pair of inequalities

LP

Maximize $c^{\top} x$
subject to

$$Ax \le b$$
$$x \ge \mathbf{0}$$

This is for the c above. Nothing to do with capacities!

1. $c_e = \begin{cases} 1 & \text{if } e \text{ out of } s \\ 0 & \text{otherwise} \end{cases}$
2. $x_e \le c(e)$
3. $\sum_{e \text{ out of } v} x_e - \sum_{e \text{ into } v} x_e \le \mathbf{0}$
4. $\sum_{e \text{ into } v} x_e - \sum_{e \text{ out of } v} x_e \le \mathbf{0}$
5. $x \ge \mathbf{0}$

Minimization or Maximization

Minimize $\boldsymbol{c}^{\top}\boldsymbol{x}$
subject to
$$\boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}$$
$$\boldsymbol{x} \geq \boldsymbol{0}$$

$\Longleftrightarrow$

Maximize $(-\boldsymbol{c})^{\top}\boldsymbol{x}$
subject to
$$(-\boldsymbol{A})\boldsymbol{x} \leq (-\boldsymbol{b})$$
$$\boldsymbol{x} \geq \boldsymbol{0}$$

different objective function value (sign flip)

Shortest Paths

Given: Directed graph $G = (V, E)$
vertices s, t in V

Find: (length of) shortest path from s to t

Claim: *Length* ℓ of the shortest path is the solution to this program.

Proof sketch: A shortest path yields a solution of cost ℓ. Optimal solution must be a combination of flows on shortest paths also cost ℓ; otherwise there is a part of the 1 unit of flow that gets counted on more than ℓ edges.

Minimize $\sum_{e} x_e$ — Sum of flow on all edges

subject to

$$x \geq 0$$

$$\sum_{e \text{ out of } s} x_e = 1$$ Flow out of s is 1

$$\sum_{e \text{ into } t} x_e = 1$$ Flow into t is 1

$$\sum_{e \text{ out of } v} x_e = \sum_{e \text{ into } v} x_e$$

for every node $v \in V - \{s, t\}$

Flow conservation

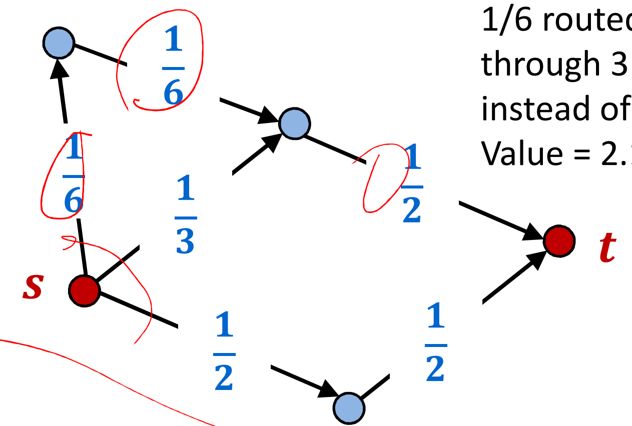
Shortest Paths

Given: Directed graph $G = (V, E)$
vertices s, t in V

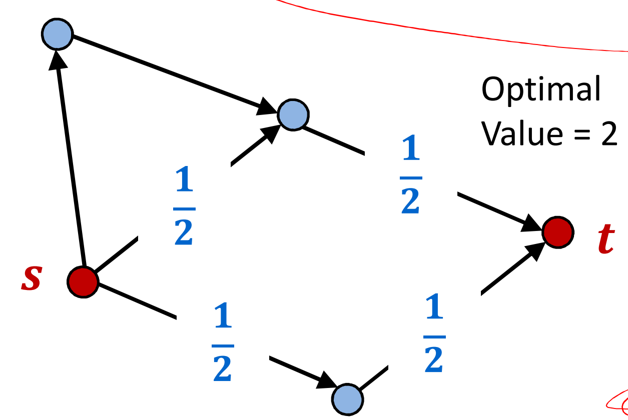
Find: shortest path from s to t

Claim: *Length* ℓ of the shortest path is the solution to this program.

Proof sketch: A shortest path yields a solution of cost ℓ. Optimal solution must be a combination of flows on shortest paths, also cost ℓ; otherwise parts of the 1 unit of flow that gets counted on more than ℓ edges.



Not optimal
1/6 routed
through 3 edges
instead of 2.
Value = 2.166666



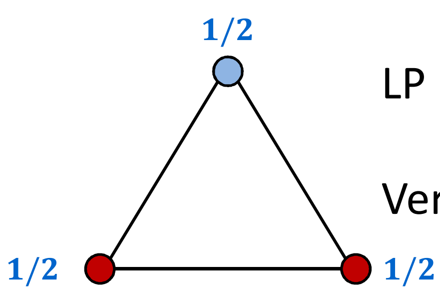
Optimal
Value = 2

Vertex Cover

Given: Undirected graph $G = (V, E)$

Find: smallest set of vertices touching all edges of G.

Doesn't work: To define a set we need $x_v = 0$ or $x_v = 1$



LP minimum $= 3/2$

Vertex Cover minimum $= 2$

Natural Variables for LP:

x_v for each $v \in V$

Minimize $\sum_v x_v$

subject to

$0 \le x_v \le 1$ for each node $v \in V$

$x_u + x_v \ge 1$ for each edge $\{u, v\} \in E$

This LP optimizes for a different problem: "**fractional vertex cover**".

x_v indicates the fraction of vertex v that is chosen in the cover.

What makes Max Flow different?

For Vertex Cover we only got a fractional optimum but for Max Flow can get integers.

- Why?
  - Ford-Fulkerson analysis tells us this for Max Flow.
  - Is there a reason we can tell just from the LP view?

Recall: Optimum is at some vertex $\boldsymbol{x}$ satisfying $\boldsymbol{A}'\boldsymbol{x} = \boldsymbol{b}'$ for some subset of exactly $\boldsymbol{n}$ constraints.

This means that $\boldsymbol{x} = (\boldsymbol{A}')^{-1}\boldsymbol{b}'$.

Entries of the matrix inverse are quotients of determinants of sub-matrices of $\boldsymbol{A}'$ so, for integer inputs, optimum is always rational.

Fact: Every full rank submatrix of MaxFlow matrix $\boldsymbol{A}$ has determinant $\pm\mathbf{1}$

$\Rightarrow$ all denominators are $\pm\mathbf{1}$ $\Rightarrow$ integers.

$\boldsymbol{A}$ is "totally unimodular"

Next: How **MaxFlow=MinCut** is an example of a general "duality" property of LPs