Lecture 20

Linear programming II

Chinmay Nirkhe | CSE 421 Spring 2025

Advertising campaign

- A political candidate has realized that if she spends \$1 on advertising on policy p she wins or loses voters according to table
- Also listed is the number of votes she needs from each demographic
- What is the minimum amount she needs to spend to win?

	Urban	Suburban	Rural
Number of votes needed	**50**	**100**	**82**
Build roads	-2	5	3
Build trains	8	-4	2
Farms	0	0	10
Gas Tax	10	0	-2

Advertising campaign

$$\min \quad x_r + x_t + x_f + x_g$$

$$\text{s.t.} \quad -2x_r + 8x_t + 0x_f + 10x_g \geq 50$$

$$5x_r - 4x_t + 0x_f + 0x_g \geq 100$$

$$3x_r + 2x_t + 10x_f - 2x_g \geq 82$$

$$x_r, x_t, x_f, x_g \geq 0.$$

	Urban	Suburban	Rural
Number of votes needed	**50**	**100**	**82**
Build roads	-2	5	3
Build trains	8	-4	2
Farms	0	0	10
Gas Tax	10	0	-2

Advertising campaign

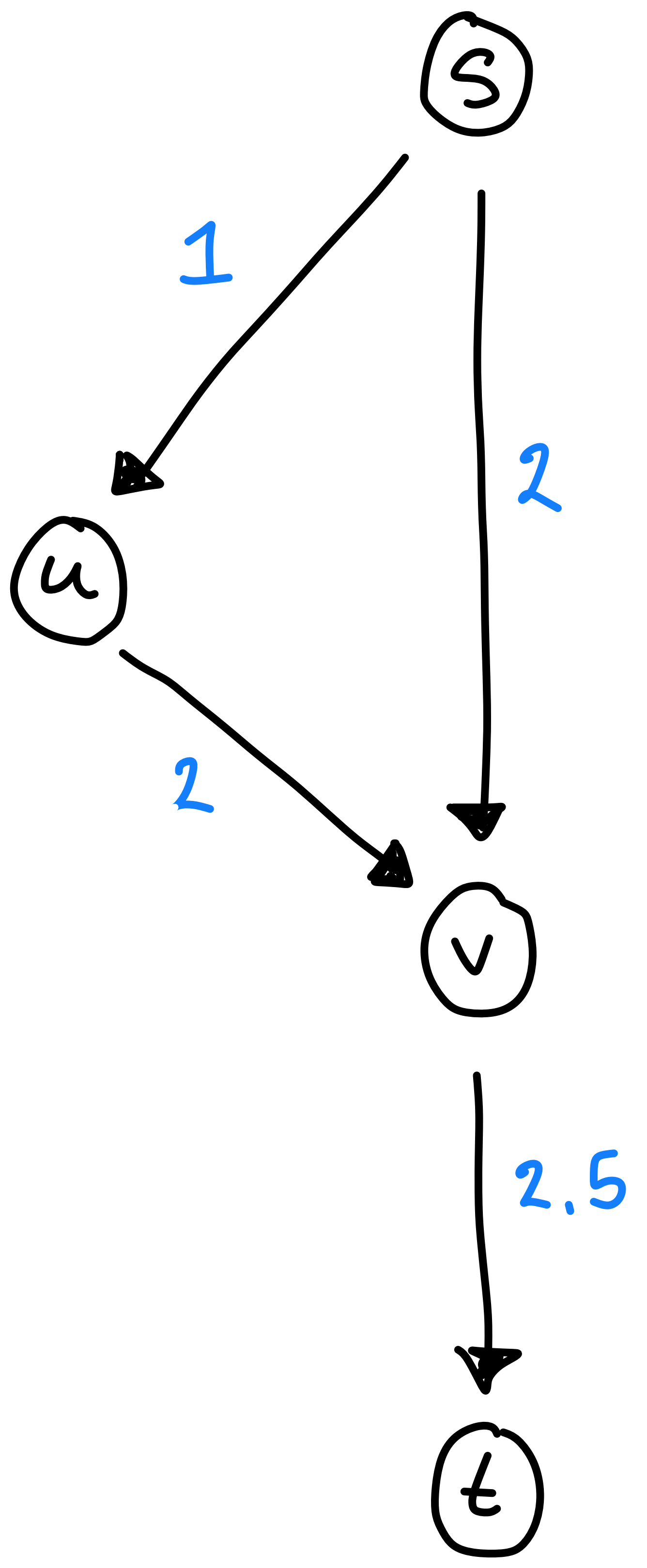
$$\min \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_r \\ x_t \\ x_f \\ x_g \end{pmatrix}$$

$$\text{s.t.} \begin{pmatrix} -2 & 8 & 0 & 10 \\ 5 & -4 & 0 & 0 \\ 3 & 2 & 10 & -2 \end{pmatrix} \begin{pmatrix} x_r \\ x_t \\ x_f \\ x_g \end{pmatrix} \geq \begin{pmatrix} 50 \\ 100 \\ 82 \end{pmatrix}$$

$$x \geq 0$$

	Urban	Suburban	Rural
Number of votes needed	**50**	**100**	**82**
Build roads	-2	5	3
Build trains	8	-4	2
Farms	0	0	10
Gas Tax	10	0	-2

Max flow as a linear program



max flow is equivalent to

$$\max \quad f_{su} + f_{sv}$$

s.t.

capacity constraints

$$\begin{aligned} 0 &\le f_{su} \le 1 \\ 0 &\le f_{sv} \le 2 \\ 0 &\le f_{uv} \le 2 \\ 0 &\le f_{vt} \le 2.5 \end{aligned}$$

conservation of flow

$$f_{su} - f_{uv} = 0$$

$$\begin{cases} f_{su} - f_{uv} \le 0 \\ -f_{su} + f_{uv} \le 0 \end{cases}$$

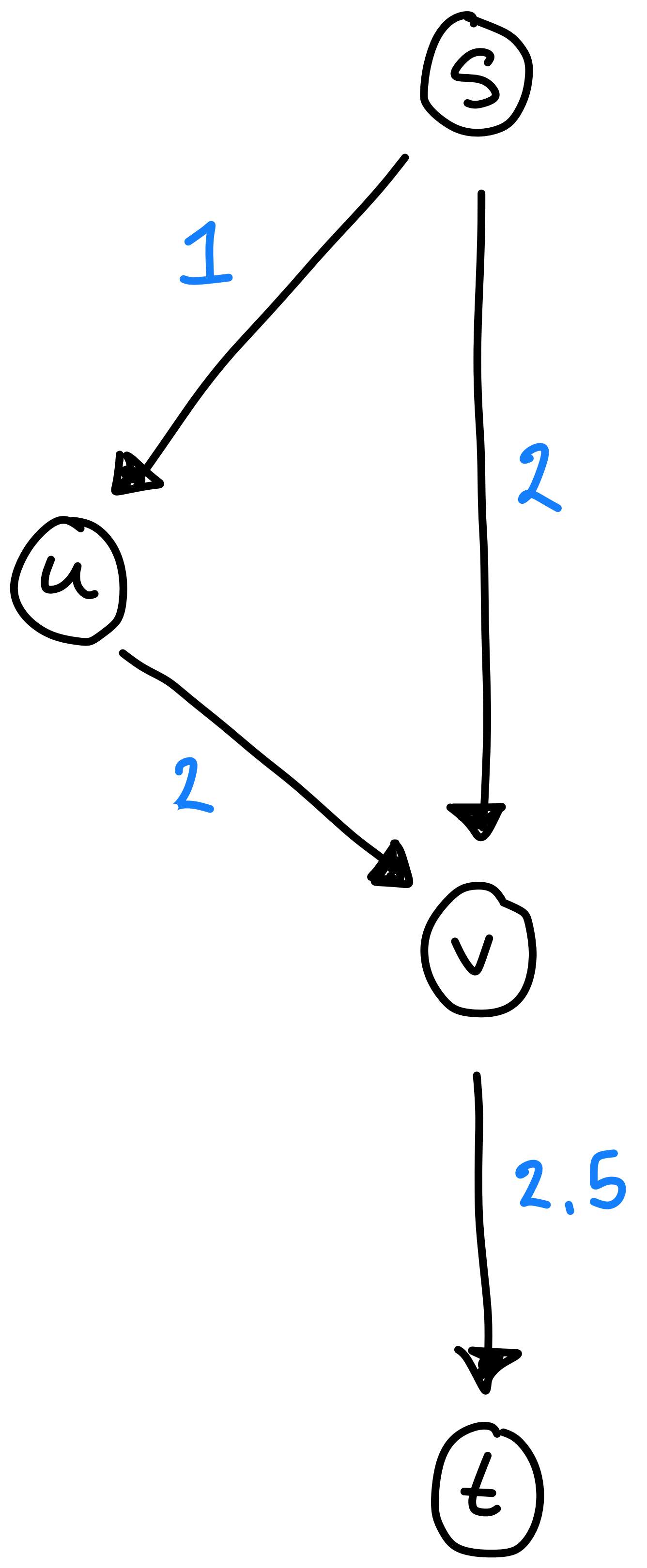
$$f_{sv} + f_{uv} - f_{vt} = 0$$

$$\begin{cases} f_{sv} + f_{uv} - f_{vt} \le 0, \\ -f_{sv} - f_{uv} + f_{vt} \le 0 \end{cases}$$

This is a linear program.

We can also express it in standard form (next slide).

Max flow as a linear program



Let $f = \begin{bmatrix} f_{su} \\ f_{sv} \\ f_{uv} \\ f_{vt} \end{bmatrix}$. Then, max flow is equivalent to:

$$\max \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \cdot f$$

$$\text{s.t.} \quad \left[\begin{array}{cccc} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \\ \hdashline 1 & 0 & -1 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 1 & 1 & -1 \\ 0 & -1 & -1 & 1 \end{array}\right] \cdot f \le \left[\begin{array}{c} 1 \\ 2 \\ 2 \\ 2.5 \\ \hdashline 0 \\ 0 \\ 0 \\ 0 \end{array}\right],$$

$$f \ge 0.$$

Max flow as a linear program

- Let (G, c, s, t) be a flow network. Then the max flow $f \in \mathbb{R}^E$ is the vector optimizing the following LP:

  - Let $g = \mathbf{1}_{\{e \text{ out of } s\}}$

  - For each vertex $v \in V \setminus \{s, t\}$, let $h_v = +\mathbf{1}_{\{e \text{ out of } v\}} - \mathbf{1}_{\{e \text{ into } v\}}$.

max flow equals

$$\max \quad g^T f$$

$$\text{s.t.} \quad \begin{bmatrix} \mathbb{I}_E \\ \hdashline h_{v_1} \\ -h_{v_1} \\ \vdots \\ h_{v_n} \\ -h_{v_n} \end{bmatrix} \cdot f \le \begin{bmatrix} c \\ \hdashline 0 \end{bmatrix}, \quad \begin{array}{l} \left.\right\} \text{capacity constraints} \\ \left.\right\} \text{conservation of flow} \end{array}$$

$$f \ge 0.$$

Max flow as a linear program

- Max flow on a graph with $|V| = n, |E| = m$ is equivalent to a linear program over m variables and $m + 2(n-2) = O(m+n)$ constraints
- If we had a very fast algorithm for solving linear programs then it would imply a very fast algorithm for max flow.
- Second, since max flow is a special case of linear programs, the algorithms we discovered for max flow may inspire algorithms for LPs.
- We will see an algorithm for LPs in next lecture.

The value of expressing problems as LPs

- Due to the prevalence of LPs, many optimizations are known
- We know LPs can be solved in polynomial time
  - Makes writing down a problem as an LP a good first step
- Writing a problem as a linear program, can make a solution apparent
- Arguing correctness of an LP can be easier
- Applying duality (next!) can give a different perspective on the problem

Minimization linear programs

$$\left\{ \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax \le b \\ & x \ge 0 \end{array} \right\} \quad \text{is equivalent to} \quad - \left\{ \begin{array}{ll} \max & (-c)^T x \\ \text{s.t.} & Ax \le b \\ & x \ge 0 \end{array} \right\}$$

Shortest paths as an LP

- **Input**: Directed graph $G = (V, E)$ and vertices s, t
- **Output**: (Length) of shortest path $s \rightsquigarrow t$
- **Claim**: The length of the shortest path is the solution to the following "flow-like" LP.
- **Proof (sketch)**:
- $(\Rightarrow)$:A path of length ℓ corresponds to a valid flow.
- $(\Leftarrow)$: A flow is the sum of $\leq m$ flows along paths. Since total flow is 1, the flow can be thought of as a probability distribution over paths. So, the LP's feasible solution is an expectation over paths.

$$\text{minimize} \quad \mathbf{1}^T \cdot f$$

$$\text{s.t.} \quad \sum_{e \text{ out of } s} f(e) = 1,$$

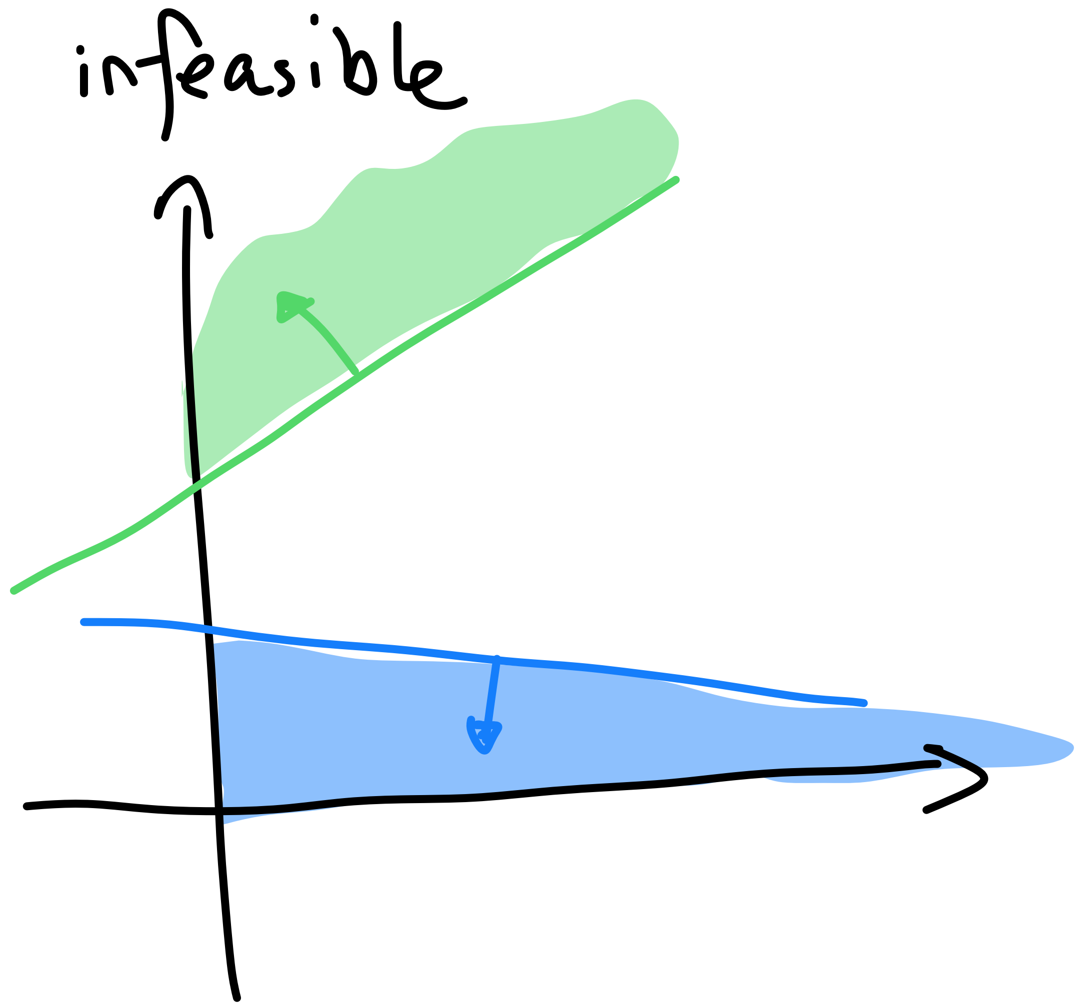
$$\sum_{e \text{ into } t} f(e) = 1,$$

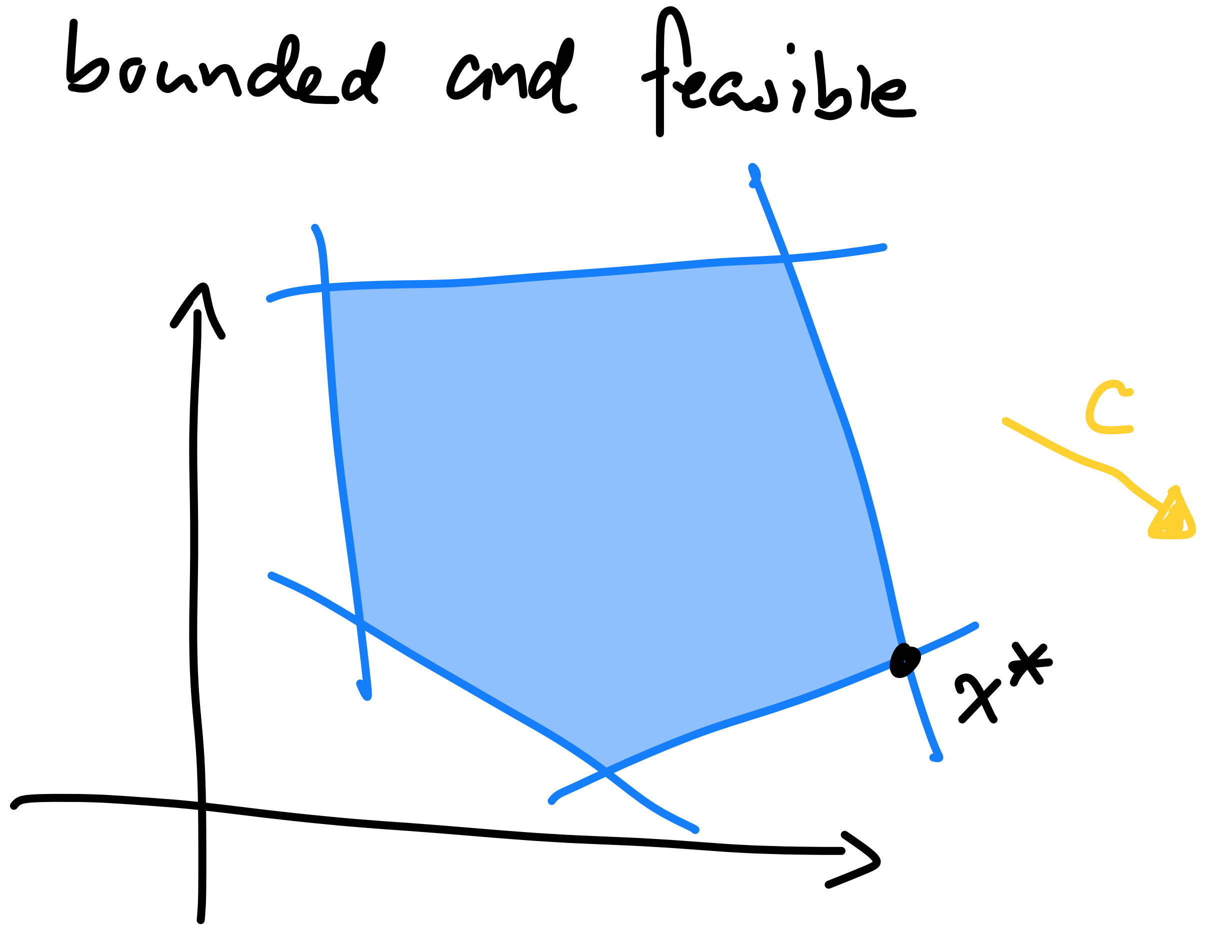
$$\forall v \in V \setminus \{s,t\}, \quad \sum_{e \text{ out of } v} f(e) = \sum_{e \text{ into } v} f(e),$$

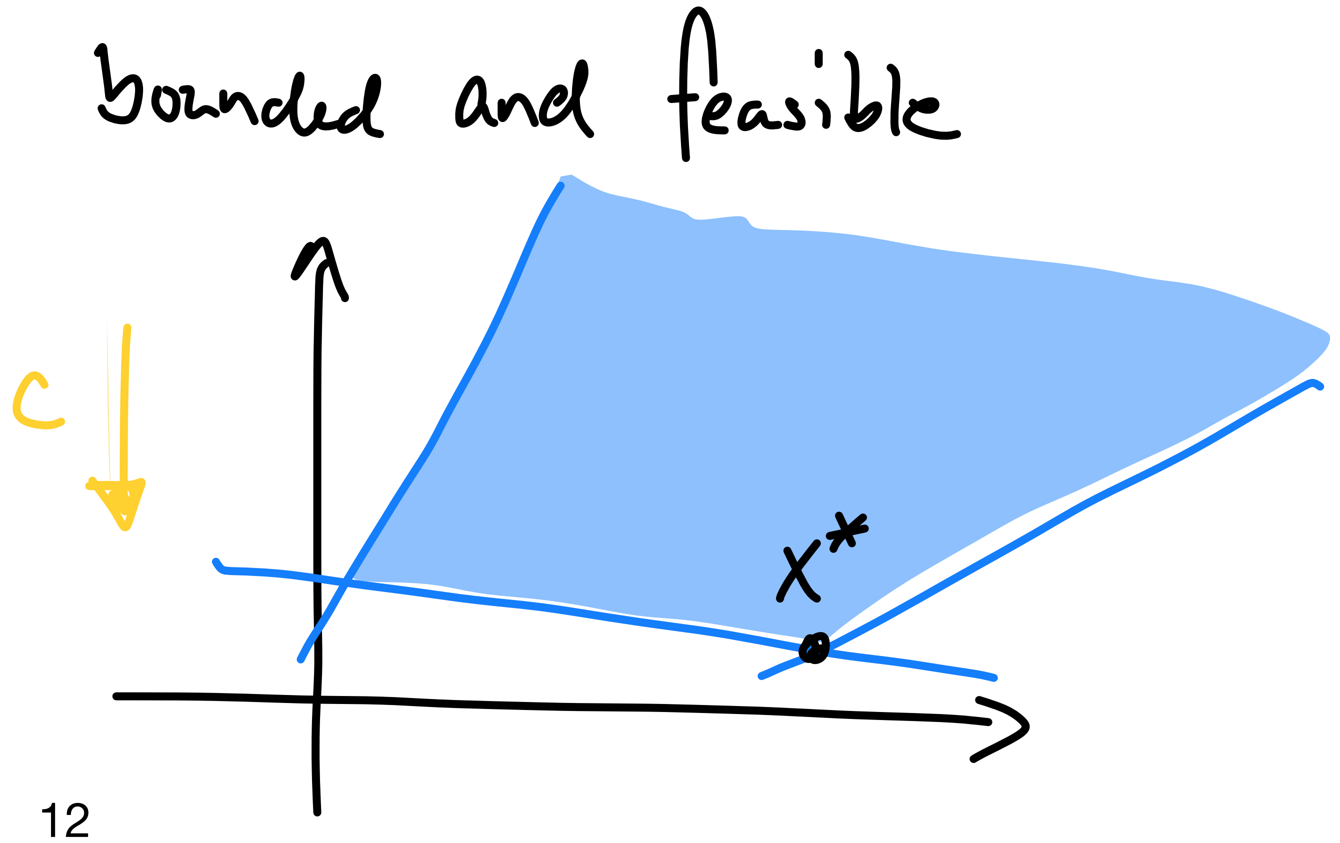
$$f \geq 0$$

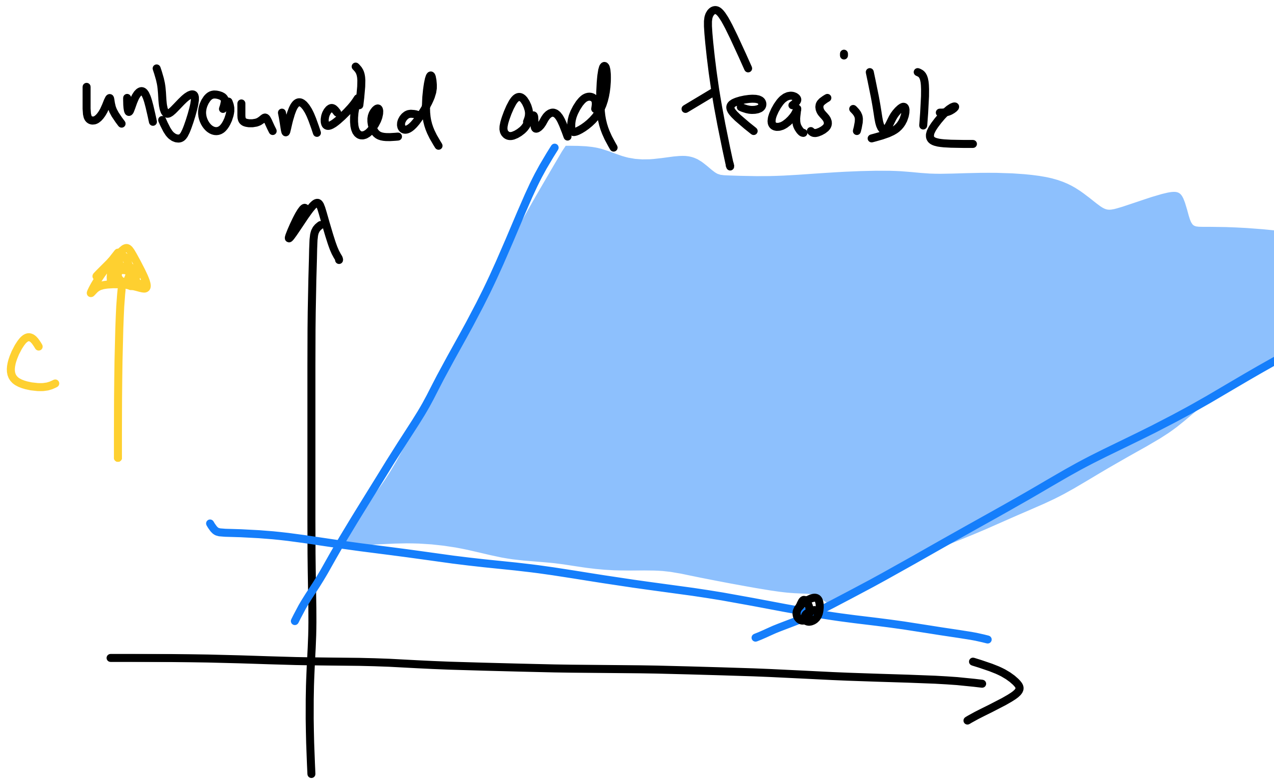
Linear programming feasibility

- Recall, the feasible region of a standard LP is $\Gamma = \{x : Ax \leq b, x \geq 0\}$.
- **Definition**: The LP is *infeasible* if $\Gamma = \emptyset$.
- **Definition**: The LP is *unbounded* if $c^\top x$ can be arbitrarily large for some $x \in \Gamma$.
- Even just deciding if a LP is feasible or not, seems like a challenging problem.



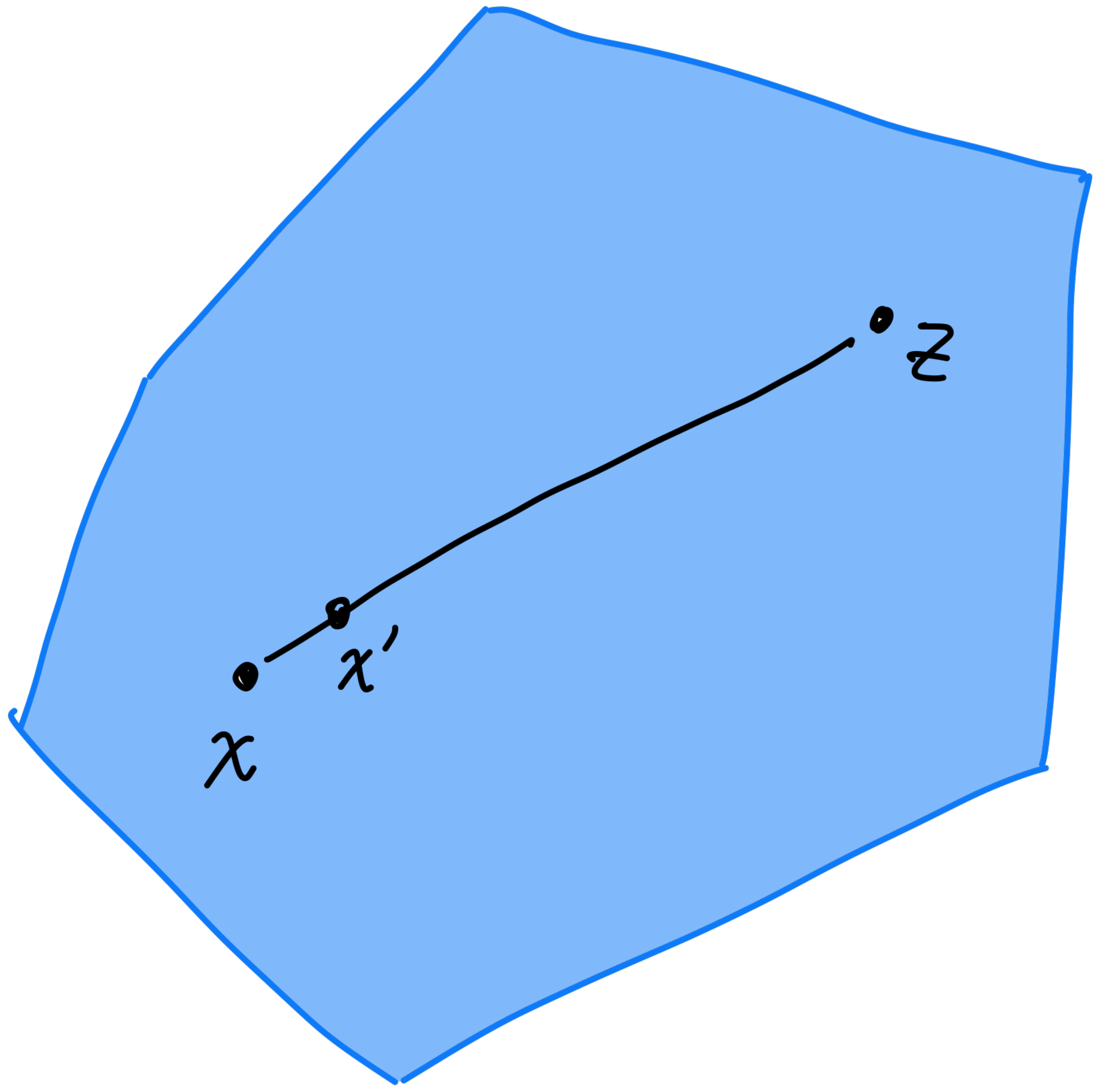






Where are the optimums of LPs

- **Theorem:** If a local optimum exists for an LP, it is a global optimum.
- **Proof:** Recall we are maximizing $c^\top x$ subject to $x \in \Gamma$ and Γ is convex. Assume x is a local optimum but not a global optimum.
  - Then $c^\top x < c^\top z$ for some $z \in \Gamma$ as x is not a global optimum.
  - Consider the line $\overline{xz} \in \Gamma$. Then $x' := x + \epsilon(z - x) \in \Gamma$ for small $\epsilon > 0$ and
  - $c^\top x' = c^\top x + \epsilon c^\top (z - x) > c^\top x.$
  - So x is not a local optimum.
  - This proves the contrapositive.

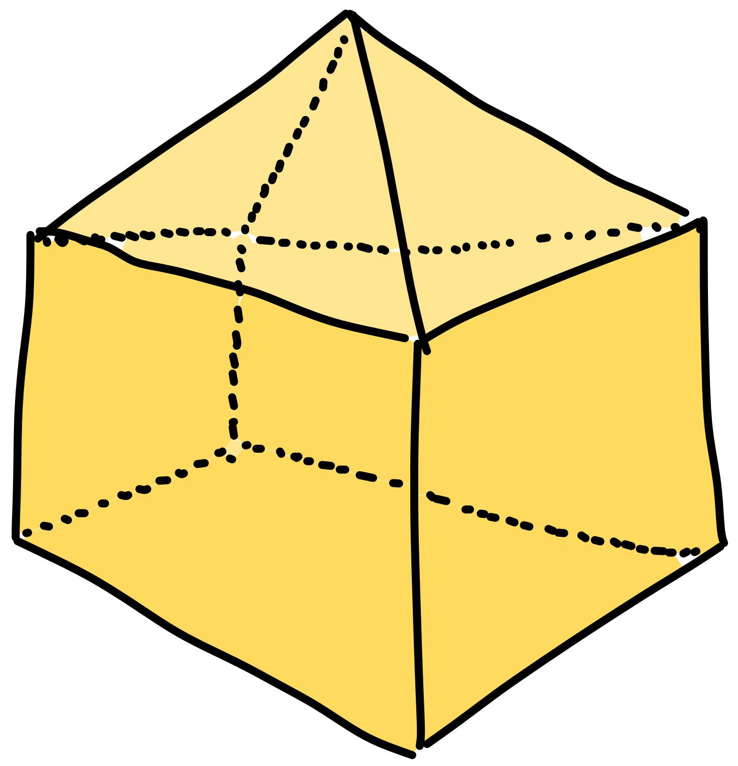


Convex polytope

- **Definition:** A **vertex** z of a convex polytope Γ is any point such that z is not the midpoint of any line segment $\overline{xy} \in \Gamma$ for $x \neq y$.

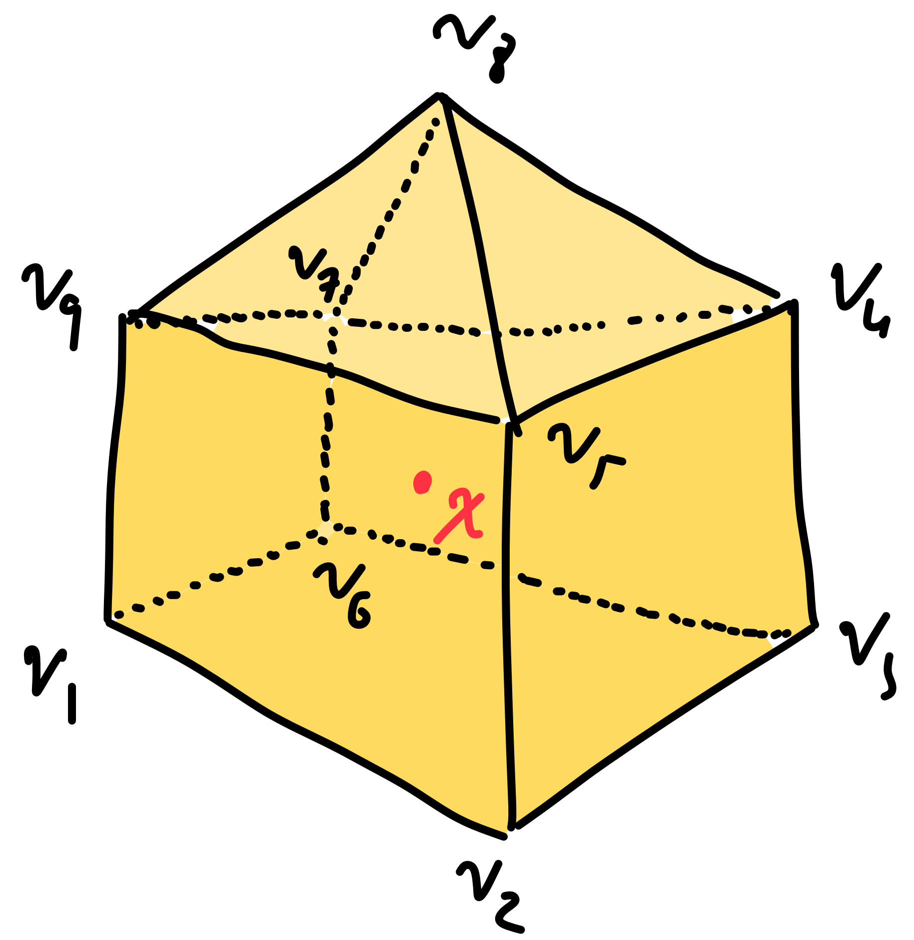
- **Remark:** If $v_1, \ldots, v_k$ are **all** the vertices of a convex polytope Γ, then $\Gamma = \mathbf{conv}(v_1, \ldots, v_k)$, the convex hull of the vertices.

- **Theorem:** If the optimum of a standard linear program is finite, then the optimum must be achieved at some vertex.



Convex polytope

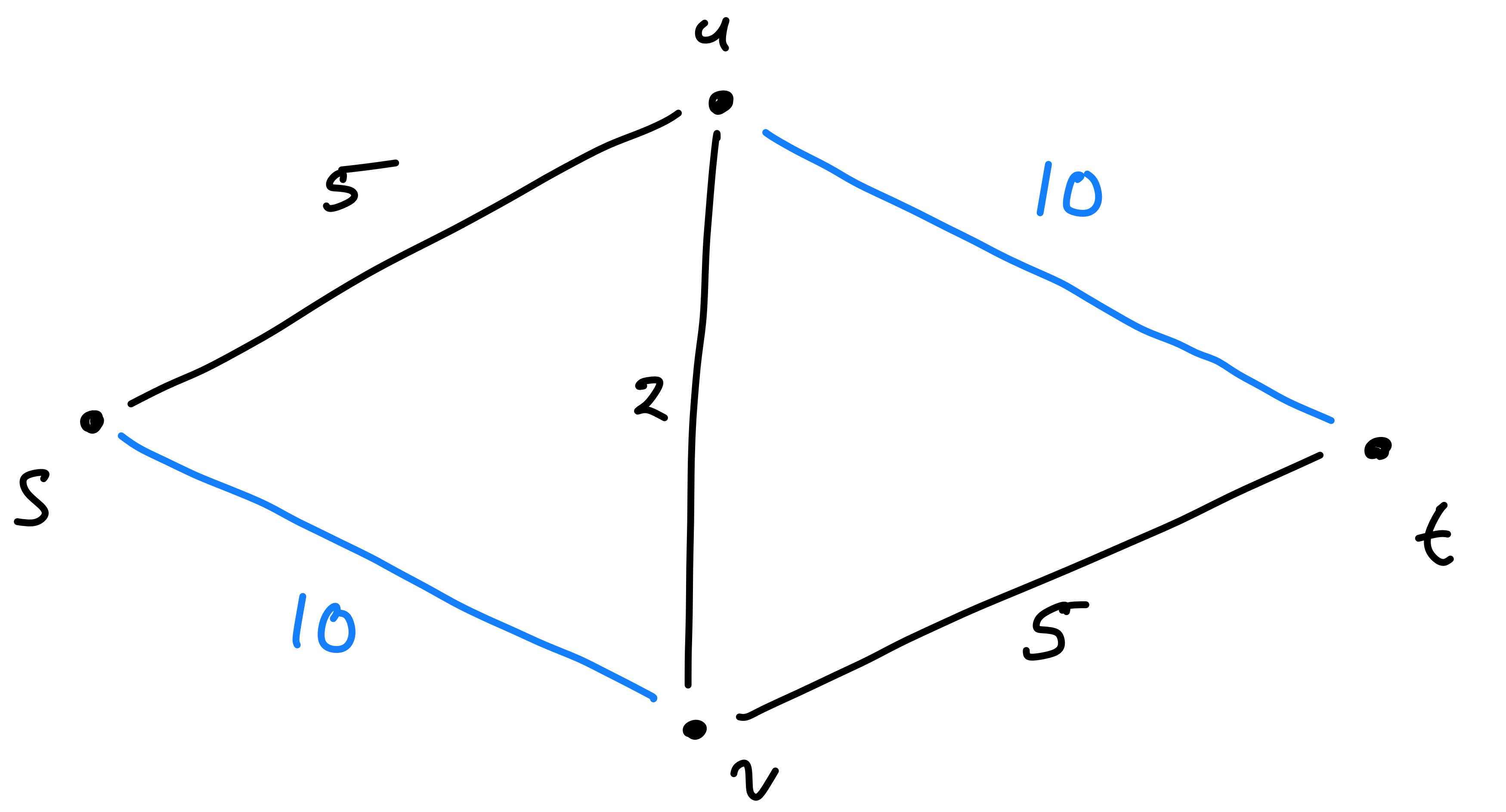
- **Theorem:** If the optimum of a standard linear program is finite, then the optimum must be achieved at some vertex.
- **Proof:** Let $v_1, \ldots, v_k$ be the vertices of the feasible region Γ.
  - Then every point $x \in \Gamma$ equals $\sum_{i=1}^{k} \lambda_i v_i$ for $\lambda \geq 0$ and $\sum_{i=1}^{k} \lambda_i = 1$.
  - By linearity of objective function,
    - $c^\top x = \sum_{i=1}^{k} \lambda_i c^\top v_i \leq \max_{i=1}^{k} c^\top v_i$
    - So one of the vertices must do better than the vertex x.

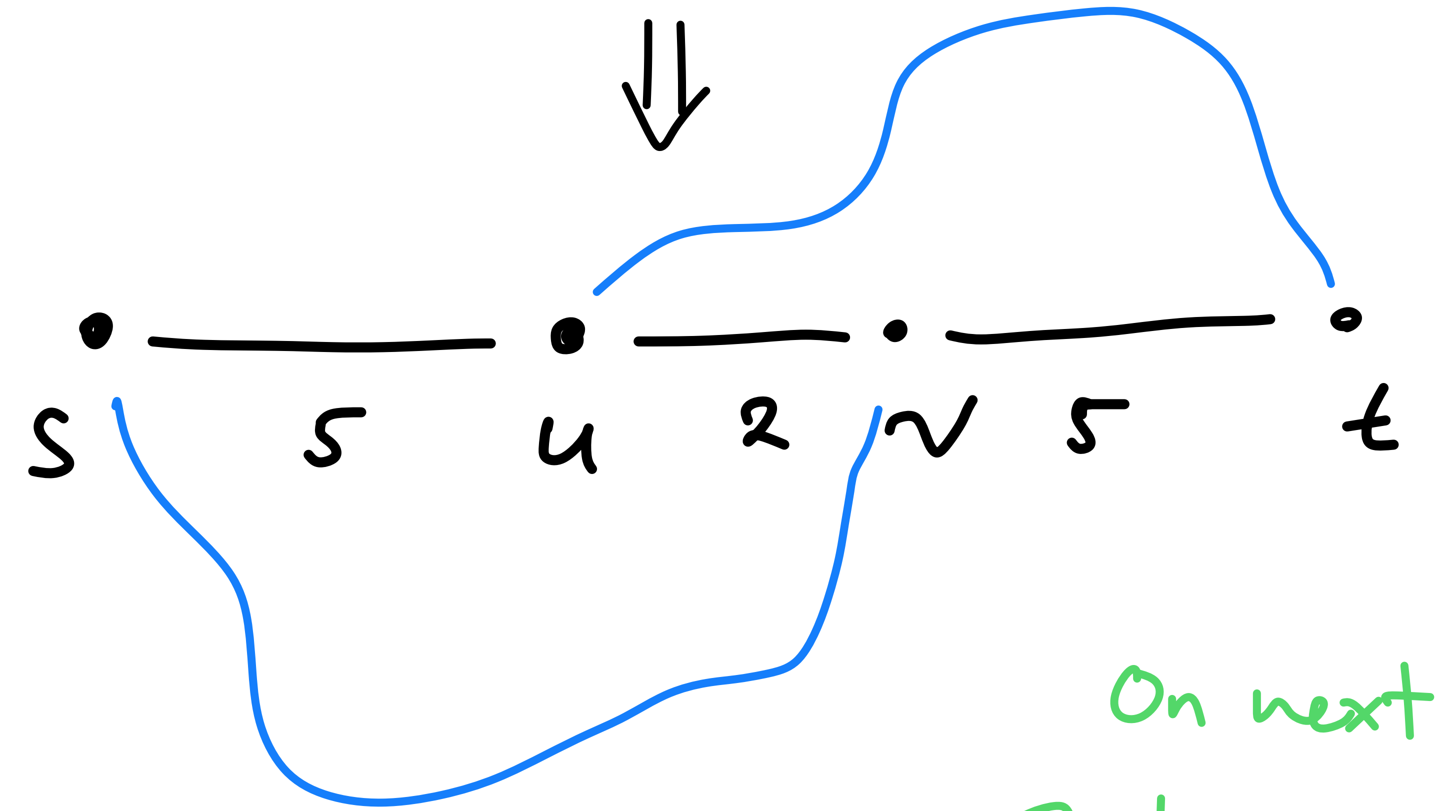


The string example

Minimization as maximization

- Recall the shortest path problem from s to t
- It is easiest seen as a *minimization* problem
- Now, imagine each edge is a piece of yarn of length $w(e)$ with knots tied at the vertices
    - Pull the yarn apart at s and t till it is taut
    - The strings that are taut form the shortest path from s to t
    - And yet pulling the yarn sounds like a *maximization* problem



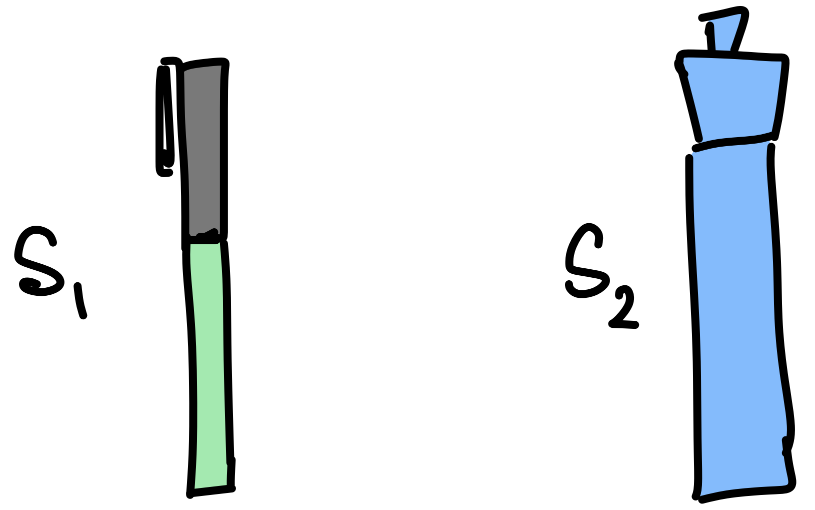


On next pset, how to convert to lin. eqs.!

$$\begin{cases} \max & |x_t - x_s| \\ \text{s.t.} & \forall e=(u,v),\ |x_u - x_v| \le d(e) \\ & x \ge 0. \end{cases}$$

Linear program duality

- Consider a salesman who sells either pens or markers.
- He sells pens for S_1 and markers for S_2.
- There are material restrictions due to labor, ink, and plastic.



$$
\begin{aligned}
\max \quad & S_1 x_1 + S_2 x_2 \\
\text{s.t.} \quad & L_1 x_1 + L_2 x_2 \leq L \\
& I_1 x_1 + I_2 x_2 \leq I \\
& P_1 x_1 + P_2 x_2 \leq P \\
& x_1, x_2 \geq 0.
\end{aligned}
$$

Linear programming duality

- Now let's imagine there are market prices for the 3 materials: y_L, y_I, y_P.
- It is only economical to **buy** a pen if $y_L L_1 + y_I I_1 + y_P P_1 \geq S_1$
  - The left hand side is the cost to make a pen **at market price**
  - And the right hand side is the cost to **buy a pen**
  - Similarly, buy markers only if $y_L L_2 + y_I I_2 + y_P P_2 \geq S_2$.
- The dual perspective is that the market **minimize** the total materials price while still being able to sell pens and markers. This is the **dual problem**.