# Complexity + Algorithms

Spring 2026

Approximation:

TSP Hitting Set

Recap:

- No reading on Thursday ←
- Next week: NP-Hardness (next week's reading)
- March 3: Midterm
  - Format: Choose 4 of 5 questions
    - → no proofs (unless I ask)
- Oral grading: signups (not) posted on Canvas. Join group, then sign up. ← tonight
- Next topic: on to graphs! (after midterm)

# Traveling Salesmen (TSP)

Given $n$ cities with pairwise distances between them, find the shortest cycle visiting all cities.

This is NP-Hard: more next week!

But idea: Take some problem X where we have reason to believe it will not have a polynomial solution.

Show any alg for TSP would be a subroutine to solve X.

So: TSP is probably hard (reduction)

Additional benefit:

The reduction for TSP is from Hamiltonian cycle:

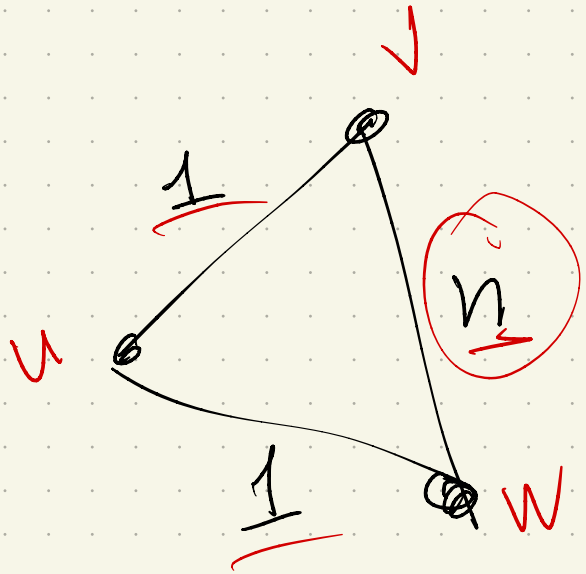
Any approximation algorithm for TSP would give an exact solution to Ham cycle

Ham cycle $\leq$ known NP-hard problem

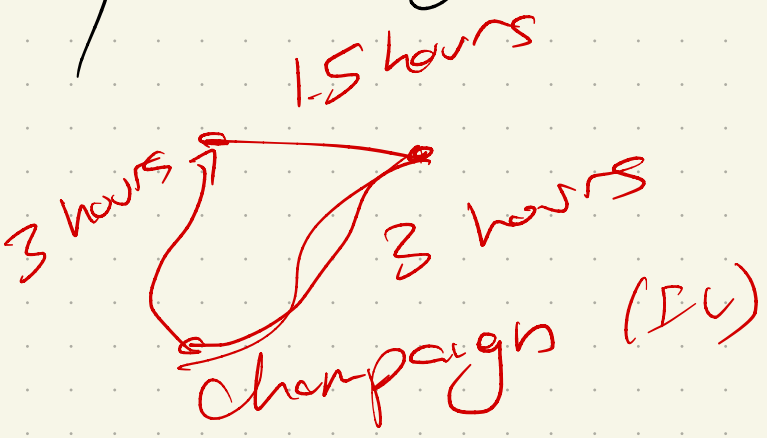
$\Rightarrow$ Hard to approximate (unless $P = NP$)

So, why study?

The reduction builds strange graphs!




Why strange?




Triangle inequality:

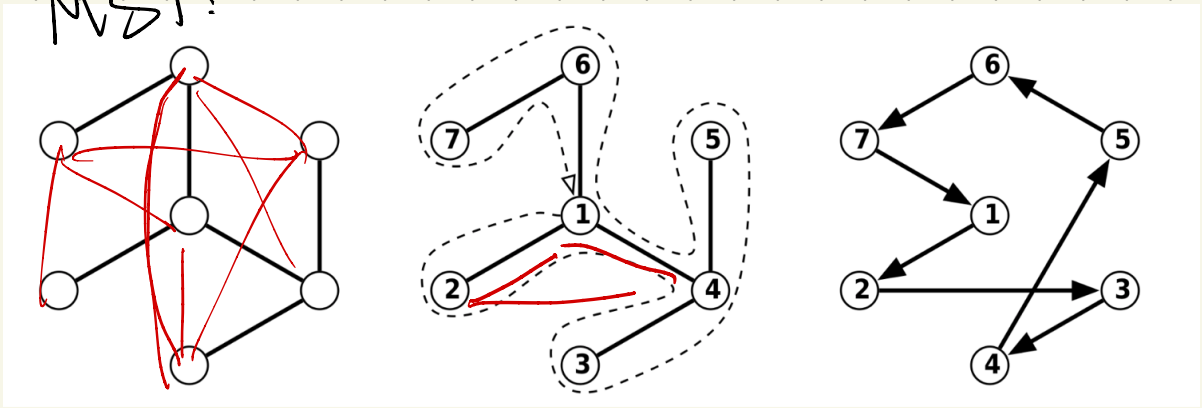
$$d(u,v) \leq d(u,w) + d(w,v)$$

common assumption

**Theorem:** If $G$ satisfies the triangle inequality, can compute a 2-approx for TSP.

Idea: Use Minimum Spanning Tree (MST):

MST:




$T$ tree of min weight that touches every $v \in V$

Then: start any where, + do tree traversal

use $\Delta$-inequality: "shortcut" while not getting too much bigger

Proof: 2-approx: $X \leq 2 \cdot OPT$

Let OPT be the cost of optimum solution to TSP.

Let MST be the weight of the Minimum spanning tree.

And let X be the weight of our constructed cycle.

Need to show:

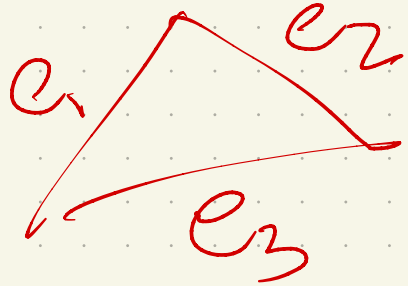
$$X \leq 2 \cdot OPT$$

Step 1: Prove $X \leq 2$ MST:

Tour uses every edge of MST exactly twice.

↳ tour $= 2$ MST

Then remove duplicate vertices on tour: use $\triangle$ inequality, only getting smaller:




$e_3 \leq e_1 + e_2$ ↳ $X$

$\Rightarrow X \leq 2 \cdot$ MST

Step 2: Prove $MST \leq OPT$:

Why? TSP solution is a cycle, so:

↳ cycle C spanning all vertices

any cycle minus an edge ⇒ path

any path is a tree


↳ this path is ~~candidate MST~~ a spanning tree

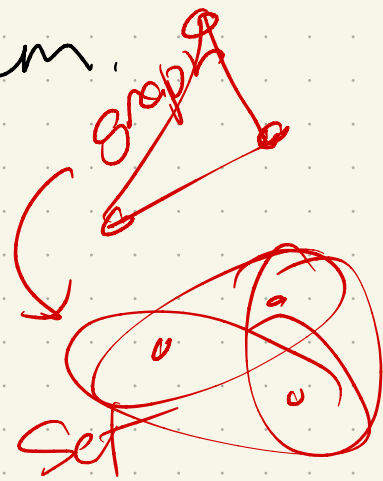
$MST \leq path = OPT - some\ edge$

$\Rightarrow MST \leq OPT.$

$1 \& 2 \Rightarrow X \leq 2MST \leq 2 \cdot OPT$ ∎

# Set Cover

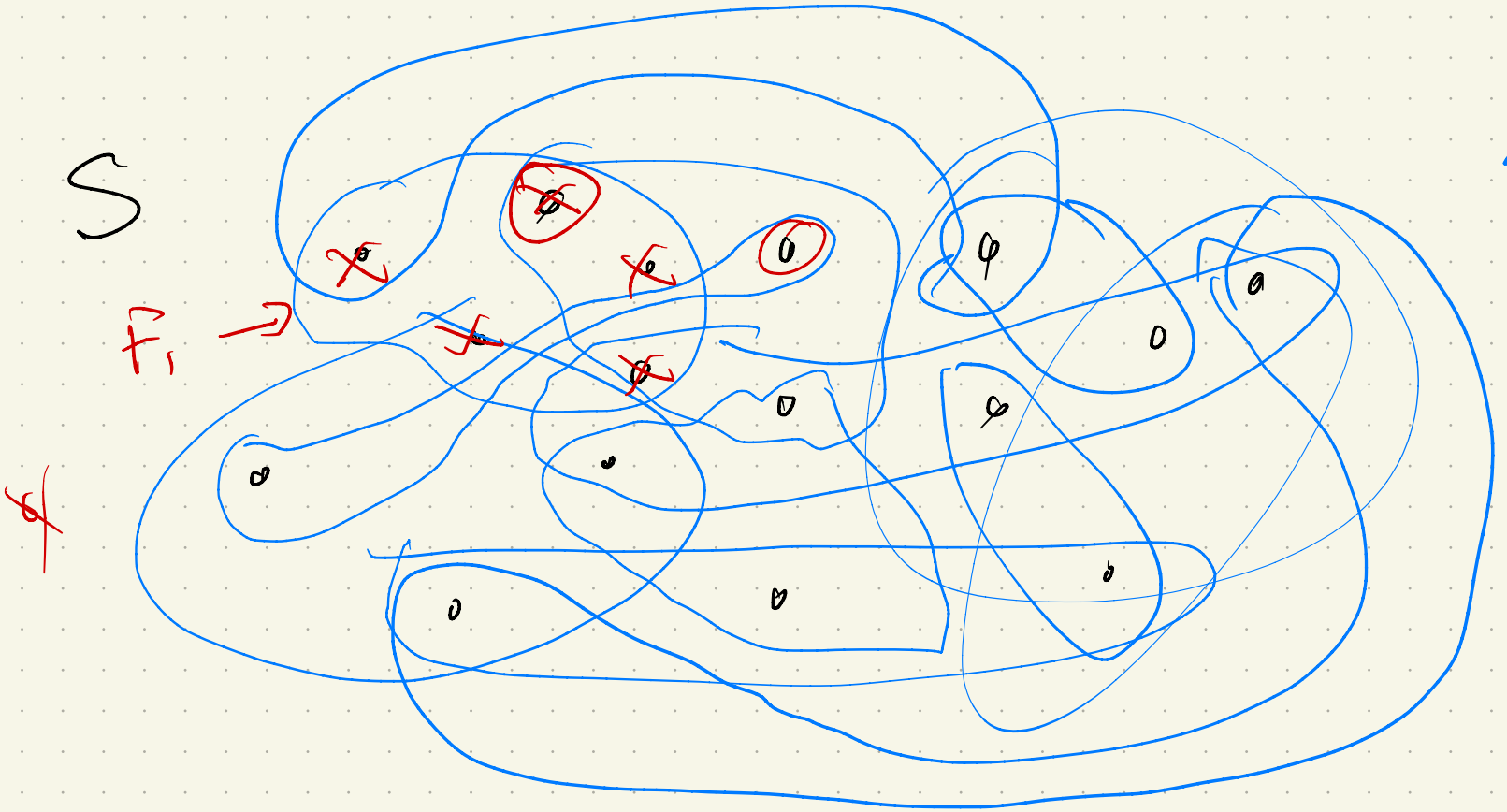
## Another classic NP-Hard Problem.




**Set Cover**

**Instance**: $(S, \mathcal{F})$:
$S$ - a set of $n$ elements
$\mathcal{F}$ - a family of subsets of $S$, s.t. $\bigcup_{X \in \mathcal{F}} X = S$.

**Question:** The set $\mathcal{X} \subseteq \mathcal{F}$ such that $\mathcal{X}$ contains as few sets as possible, and $\mathcal{X}$ covers $S$. Formally, $\bigcup_{X \in \mathcal{X}} X = S$.



# Greedy Set Cover

How should we be greedy?

$i \leftarrow 1$

while $S \neq \phi$

take largest set in $\mathcal{F}$

↳ $F_i$

delete elements in $F_i$

$i \leftarrow i+1$

# Sanity Check: does it work?

does T wind up empty?

- yes → eventually could use all of $\mathcal{F}$ to cover all of S

```
GreedySetCover(S, F)
    X ← ∅;  T ← S
    while T is not empty do
        U ← set in F covering largest
            # of elements in T
        X ← X ∪ {U}
        T ← T \ U

    return X.
```

# rep = |X|

Observation: Let $\alpha_i$ be # of new elements covered in iteration $i$ of loop.

Then, $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m$, where $m$ is total # of iterations.

Why? Because otherwise would set larger one first!

Notation:

Let $\{V_1, \ldots, V_k\} \subseteq \mathcal{F}$ be OPT set cover.

Let $T_i$ = uncovered elements at beginning of iteration $i$

and $U_i$ = set chosen in $i^{th}$ iteration

$T_1 \to T_1 - \alpha_1 = T_2 \to T_2 - \alpha_2 = T_3 \ldots$

```
GreedySetCover(S, F)
    X ← ∅;  T ← S
    while T is not empty do
        U ← set in F covering largest
            # of elements in T
        X ← X ∪ {U}
        T ← T \ U

    return X.
```

$|U_1| = \alpha_1$

$|U_2| = \alpha_2$

$|U_3| = \alpha_3$

$X$ = set $U_1, U_2, \ldots U_\ell$

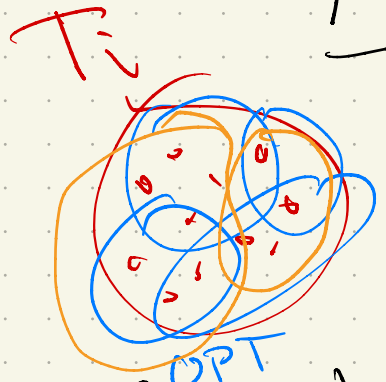
$\ell$ = # loop repetitions

Compare $k$ & $\ell$

**Lemma:** $a_i \geq \frac{|T_i|}{k}$ ← # elements left at beginning of $i$th loop; # elements removed in $i$th loop; $k \leftarrow |OPT|$

Proof: Consider OPT again → $k$ sets, $\forall$ covers $T_i$.

Some set in OPT must have size $\frac{|T_i|}{k}$ → why? all under average then sum is too small




← $|T_i| = 11$, $k = 4$ if all $\leq 2$, $2+2+2+2 < 11$

Greedy picks biggest coverage, so

$|U_i| \geq \frac{|T_i|}{k}$ since the set is OPT is candidate

Rewrite: if $\alpha_i \geq \frac{|T_i|}{k}$ and

$$|T_{i+1}| = |T_i| - \alpha_i$$

fact (1)

$$\Rightarrow |T_{i+1}| \leq |T_i| - \frac{|T_i|}{k} = \left(1 - \frac{1}{k}\right)|T_i|$$

Thm: GreedySet cover is $O(\log n)$ approx.

Proof: Need to know how many times the loop runs (since adds 1 set per iteration)

GreedySetCover$(S, \mathcal{F})$
  $\mathcal{X} \leftarrow \emptyset$; $T \leftarrow S$
  **while** $T$ is not empty **do**
    $U \leftarrow$ set in $\mathcal{F}$ covering largest # of elements in $T$
    $\mathcal{X} \leftarrow \mathcal{X} \cup \{U\}$
    $T \leftarrow T \setminus U$
  **return** $\mathcal{X}$.

Well, fact #1

$$|T_i| \le \left(1-\frac{1}{k}\right)|T_{i-1}| \le \left(1-\frac{1}{k}\right)^2 |T_{i-2}|$$

Stop when $|T_i|=0$ $\le \cdots \left(1-\frac{1}{k}\right)^i |T| = \left(1-\frac{1}{k}\right)^i \cdot n$

When do we reach an iteration $M$ where this bound shows $|T_M| < 1$? (Loop would then stop).

Math tricks: $1-x < e^{-x}$ for $x \ge 0$.

Let $M = \lceil 2k \ln n \rceil$:

$$|T_M| \le \left(1-\frac{1}{k}\right)^M \cdot n > \left(1-\frac{1}{k}\right)^{\lceil 2k \ln n \rceil} n$$

(Math, cont):

$$|T_M| \leq \left(1-\frac{1}{k}\right)^M \cdot n = \left(1-\frac{1}{k}\right)^{\lceil 2k \ln n \rceil} \cdot n$$

$$\leq \left(e^{-\frac{1}{k}}\right)^{\lceil 2k \ln n \rceil} \cdot n$$

(b/c $1-x < e^{-x}$, let $x = \frac{1}{k}$)

$$\leq e^{-\frac{2k \ln n}{k}} \cdot n \quad (\text{b/c } (y^a)^b = y^{ab})$$

$$= e^{-2\ln n} \cdot n = \left(e^{\ln n}\right)^{-2} \cdot n$$

$$= \left(n^{-2}\right) \cdot n = \frac{n}{n^2} = \frac{1}{n} < 1$$

∎

End recap: If $|OPT| = k$:

After $2k \ln n$ repetitions, $T$ is empty.

```
GreedySetCover(S, F)
    X ← ∅;  T ← S
    while T is not empty do
        U ← set in F covering largest
            # of elements in T
        X ← X ∪ {U}
        T ← T \ U

    return X.
```

So: $|X| \leq \lceil 2k \ln n \rceil$

$$\Rightarrow \frac{|X|}{|OPT|} \leq \frac{\lceil 2k \ln n \rceil}{k} = O(\log n)$$

$\Rightarrow O(\log n)$ approximation.

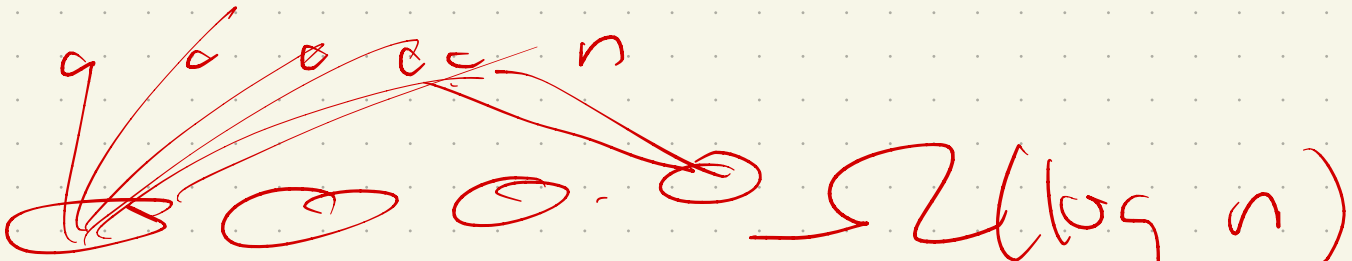
upper bound

Lower bound:

Take vertex cover

↳ Set cover problem

- Make each edge an element in $S$
- Create a set in $\mathcal{F}$ for every vertex, with all adj edges in set


$\Omega(\log n)$

To show lower bound:
build instance of sets s.t.
greedy algorithm has size $\geq \lg n \cdot OPT$

↳ build sets + show ineq.

n "left" sets
Right sets: $U_2, \ldots, U_n$
↳ make some elements
s.t. they belong to certain sets

# Clustering

**Input:** $X \subseteq \mathbb{R}^d$, $X = \{x_1, \dots, x_n\}$ + distance

$$d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$$

**Output:** A set of clusters $X_1, \dots, X_k$, each associated with a point $c_i \in \mathbb{R}^d$: $\Phi : X \to \{c_1, \dots, c_k\}$

**Cost of cluster:**

$k$-means: $\min \sum_{x \in X}$

$k$-center: $\min \{ \max$

$k$-median:

# K-center

Bad news: NP-Hard

Even to approximate within a factor of $c < 2$

How?

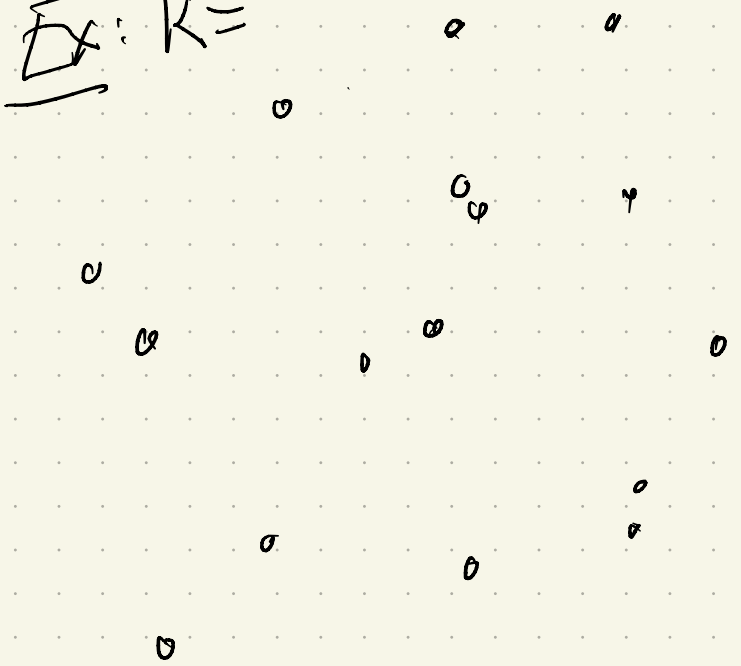
Show a better than 2-approx would let us solve some NP-Hard problem.

# K-center: Gonzalez's algorithm

Be greedy!

- Pick random point + make $c_1$
- While we don't have k points
  choose furthest point away

Ex: k=



Formally

```
GonzalezKCenter(P, k):
  for i ← 1 to n
      d_i ← ∞
  c_1 ← p_1
  for j ← 1 to k
      r_j ← 0
      for i ← 1 to n
          d_i ← min{d_i, |p_i c_j|}
          if r_j < d_i
              r_j ← d_i;  c_{j+1} ← p_i
  return {c_1, c_2, ..., c_k}
```