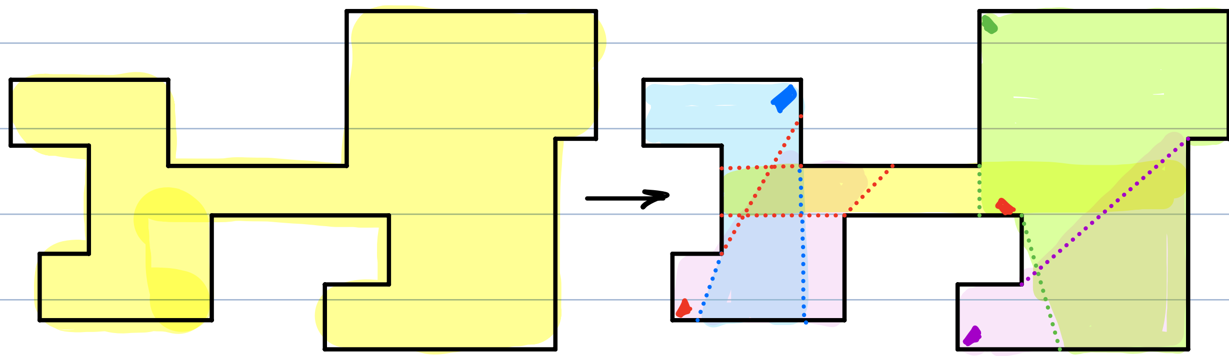
CMSC 451 - Algorithm Design
Lecture 7 - Greedy Approximation: Set Cover

Set Cover - Cover a domain with a minimum number of sets

Applications:

Surveillance - Place a min. number of cameras in art gallery.



Wireless/Cellular Coverage - Place min. number of wireless routers/cell towers to cover some region.

Workforce Scheduling - Given when/where employees can work, schedule to cover all times/locations.

Set System: $\Sigma = (X, \mathcal{S})$ where:

$X = \{x_1, \dots, x_n\}$ domain of things to be covered.

$\mathcal{S} = \{s_1, \dots, s_m\}$ collection of sets that can be used to build the cover.

Throughout: $n = |X|$, $m = |\mathcal{S}|$

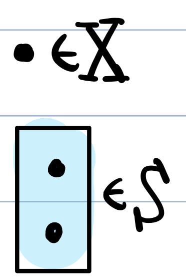
We assume: $s_1 \cup s_2 \cup \dots \cup s_m = X$ - $\mathcal{S}$ covers X.

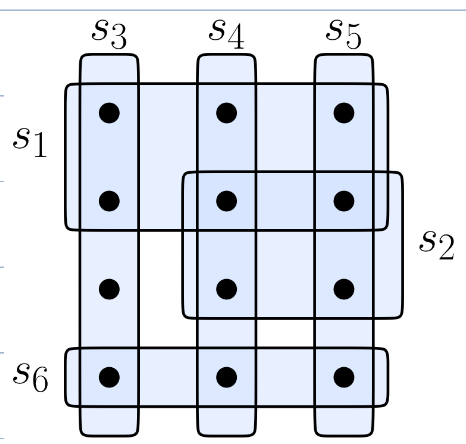
Set Cover Problem - Given a set system $(X, \mathcal{S})$ find smallest number of sets that cover all elements of X.

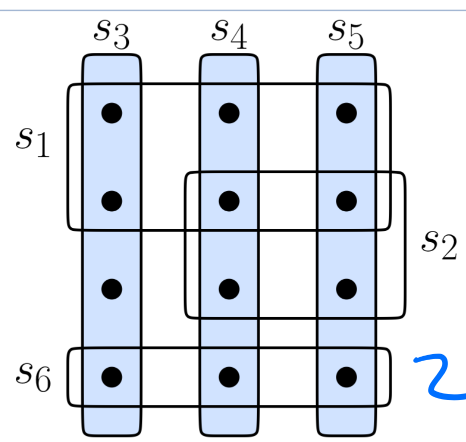
fewest num. of sets

i.e., Find $C \subseteq \{1, \dots, m\}$ of min. size s.t.

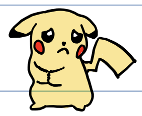
$$\bigcup_{i \in C} s_i = X$$ ← cover entire domain







$\hookrightarrow C = \{3,4,5\}$



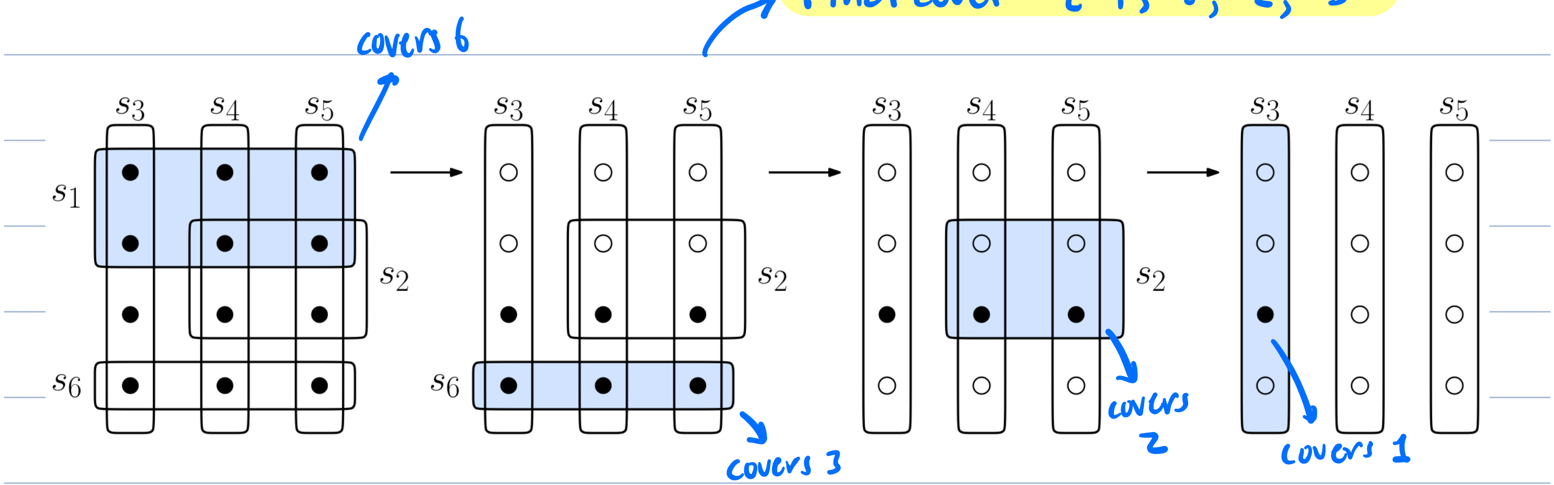
Bad news: Set cover is NP-hard.

We'll prove greedy heuristic gives $(\ln |X|)$ approx.

Greedy Heuristic - Repeatedly add the set that covers the most uncovered items.

```
greedy-set-cover (X, S) {
    U ← X                              // U = uncovered items
    C ← ∅                              // C = indices of cover
    while (U ≠ ∅)
        i ← set s_i covers most elts of U
        add i to C
        U ← U \ s_i                    // remove covered items
                                          (set subtraction)
    return C
```

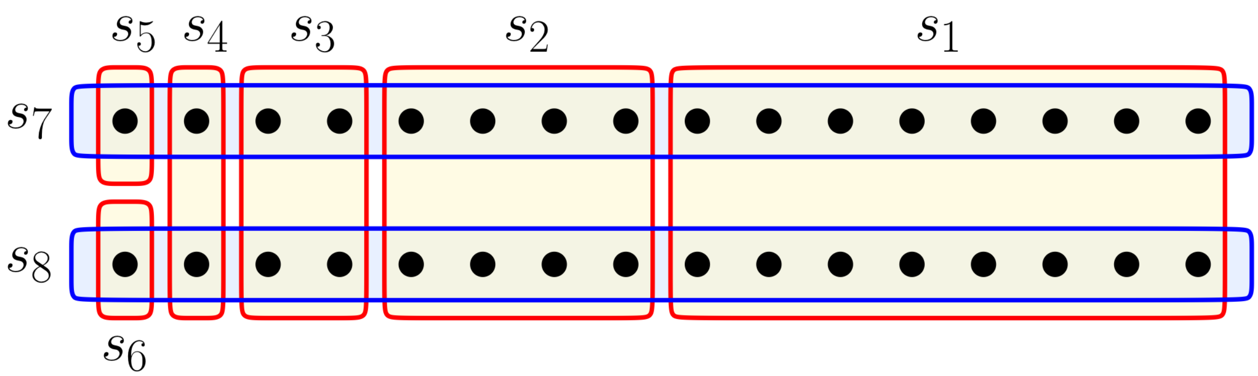
Final cover - $\{s_1, s_6, s_2, s_3\}$



Running Time: $O(n \cdot m)$ $\quad n = |X| \quad m = |S|$

(see pdf notes for more info)

Bad Case for Greedy:



$|X| = 32 = 2^5$

Opt cover $= \{s_7, s_8\}$ - 2 sets

Greedy: s_1, s_7, s_8 all cover 16 → take s_1

s_2, s_7, s_8 all cover 8 → take s_2

s_3, s_7, s_8 all cover 4 → take s_3

⋮ 〃 s_4, s_5, s_6

Greedy takes 6 sets

We can generalize this to $|X| = n = 2^k$

Opt (O) takes 2 sets. Greedy (G) takes $k+1$.

Approximation ratio $= \dfrac{|G|}{|O|} \approx \dfrac{\lg n}{2} = O(\log n)$

$\lg \equiv \log_2$

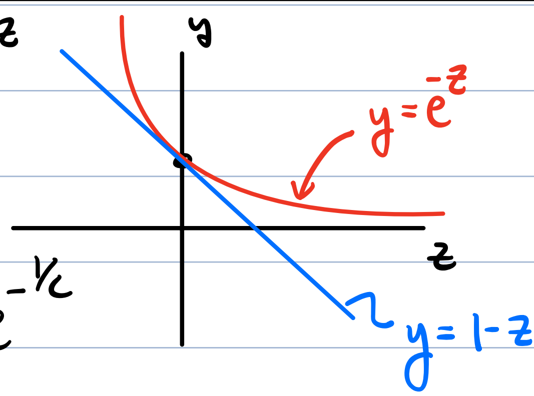
We'll show that approx. ratio $\leq \ln n = \ln |X|$
for any set system $(X, \mathcal{S})$

> **Thm:** Given any set system $(X, \mathcal{S})$, let
> O = opt set cover, G = greedy set cover, then
> $$\frac{|G|}{|O|} \leq \ln |X|$$

Proof: Utility lemma –

> **Lem:** $\forall c > 0$,
> $$(1 - 1/c) \leq e^{-1/c}$$
> base of ln
> $e \approx 2.718\ldots$

Pf: Fact: $\forall z,\ 1 - z \leq e^{-z}$



Setting $z = 1/c$
$\Rightarrow\ 1 - 1/c \leq e^{-1/c}$

Let $o = |O|$, $g = |G| - 1$ cheat! (Simplifies math)

We'll show $g \leq o \cdot \ln n$, $n = |X|$.

Notation: Let X_i = items that remain uncovered at end of i^{th} iteration

$n_i = |X_i|$ $(= |U|$ in algorithm$)$

$(n_0 = n)$

Observations:

- At start of stage i, n_{i-1} items remain
- $\exists$ cover of X of size σ
  - $\Rightarrow \exists$ cover of X_{i-1} of size σ
  - $\Rightarrow \exists$ set covers $\geq n_{i-1}/\sigma$ items of X_{i-1}

  (pigeonhole principle)

- Greedy covers at least this many

$$\Rightarrow \quad n_i \leq n_{i-1} - (n_{i-1}/\sigma)$$

$$= n_{i-1}(1 - 1/\sigma)$$

Each stage reduces no. remaining by $*(1-1/\sigma)$

- By induction:

$$n_i \leq n_{i-1}(1-1/\sigma) \leq n_{i-2}(1-1/\sigma)^2 \leq n_{i-3}(1-1/\sigma)^3$$

$$\leq n(1-1/\sigma)^i$$

Recall $g = |G| - 1$

- After iteration g, still have ≥ 1 item remain.

$$\Rightarrow \quad 1 \leq n_g \leq n(1-1/\sigma)^g$$

- By utility lemma $(1-1/\sigma) \le e^{-1/\sigma}$

$$\Rightarrow \quad 1 \le n\left(e^{-1/\sigma}\right)^g = n \cdot e^{-g/\sigma} = n/e^{g/\sigma}$$

$$\Rightarrow \quad e^{g/\sigma} \le n$$

$$\Rightarrow \quad g/\sigma \le \ln n \qquad \text{(take ln both sides)}$$

$$\Rightarrow \quad \frac{|G|-1}{|\sigma|} \le \ln n \qquad \text{since } g = |G|-1 \text{ + } \sigma = |\sigma|$$

Ignoring the "-1", this is what we want! $\square$

How bad is this?

- Ideally approx. ratio can be made arbitrarily small – $(1+\varepsilon)\cdot \mathrm{Opt}$
- Next best – Approx ratio is a small constant e.g., $2\cdot\mathrm{Opt}$, $3\cdot\mathrm{Opt}$
- For set cover, approx ratio grows (slowly) with domain size – $(\ln |X|)\cdot\mathrm{Opt}$

- Hope for better? No

  Can't beat $O(\log n)$ unless $P = NP$.