Set-Covering

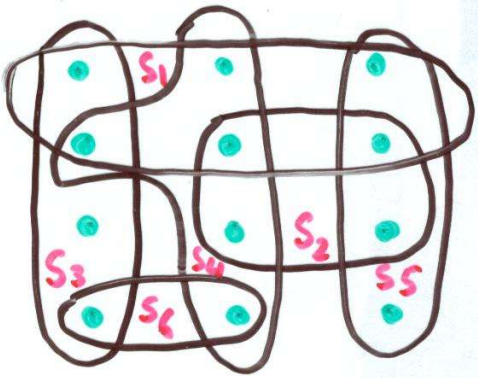
X: a finite set.

F: family of subsets of X.

(X, F) covers X if $X = \bigcup_{S \in F} S$.

The problem is to minimize the number of subsets in a collection

$C \subseteq F$ s.t. $X = \bigcup_{S \in C} S$.



X has 12 green dots.

$F = \{S_1, S_2, S_3, S_4, S_5, S_6\}$.

A minimum set cover is $C = \{S_3, S_4, S_5\}$.

A greedy algorithm.

```
Greedy-Set-cover(X, F)
  U := X
  C := ∅
  while U ≠ ∅ do
      select an S ∈ F that maximizes |S ∩ U|
      U := U - S;
      C := C ∪ S;
  endwhile
  return C.
```

The algorithm chooses the set which covers most elements from the uncovered ones so far.

The algorithm runs in polynomial time.
A crude implementation runs in $O(|X| \cdot |F| \cdot \min(|X|, |F|))$ time since there are $O(\min(|X|, |F|))$ iterations in while loop each can be implemented in $O(|X| \cdot |F|)$ time easily.

Let $H(d) = \sum_{i=1}^{d} 1/i$ and $H(0) = 0$.

Theorem. Greedy-Set-cover is a polynomial-time $\rho(n)$-approximation algorithm, where

$$\rho(n) = H(\max\{|S| : S \in F\}).$$

Proof. Assign a cost of 1 to set selected by the algorithm and distribute the cost evenly among all elements that are covered for the first time.

C^*: optimal set cover

C: set cover computed by the algorithm.

S_i: ith subset selected by the algorithm.

c_x: cost assigned to $x \in X$.

if x is covered for the first time when S_i is selected, then

- $c_x = \dfrac{1}{|S_i - (S_1 \cup S_2 \cup \cdots \cup S_{i-1})|}$.

① - $|C| = \sum_{x \in X} c_x$ since 1 unit cost is assigned for each selected set.

- The cost for optimal cover is

$$\sum_{S \in C^*} \sum_{x \in S} c_x$$

- Since each $x \in X$ is in at least one set $S \in C^*$, we have

② $$\sum_{S \in C^*} \sum_{x \in S} c_x \geq \sum_{x \in X} c_x .$$

- From ① & ②

$$|C| \leq \sum_{S \in C^*} \sum_{x \in S} c_x .$$

③ - We prove that $\sum_{x \in S} c_x \leq H(|S|)$.

- From ② & ③ we get

$$|C| \le \sum_{S \in C^*} H(|S|)$$

$$\le |C^*| \cdot H(\max\{|S| : S \in F\}).$$

Now prove ③: Consider any $S \in F$.

- Let $u_i = |S - (S_1 \cup S_2 \cup \cdots \cup S_i)|$ for $i = 1, 2, \ldots, |C|$.

be the number of elements in S still not covered by $S_1, S_2, \cdots S_i$.

$u_0 = |S|$.

- Let k be the smallest index s.t. $u_k = 0$, that is, all elements of S are covered by $S_1, S_2, \cdots S_k$.

- $u_{i-1} \ge u_i$ and $u_{i-1} - u_i$ is the # elements in S covered first time by S_i.

Thus,

$$\sum_{x \in S} c_x = \sum_{i=1}^{k} (u_{i-1} - u_i) \cdot \frac{1}{|S_i - (S_1 \cup S_2 \cdots \cup S_{i-1})|}. \quad (4)$$

- Observe that

$$|S_i - (S_1 \cup S_2 \cdots \cup S_{i-1})| \geq |S - (S_1 \cup S_2 \cup \cdots \cup S_{i-1})|$$

since greedy choice selected S_i instead of S. $= u_{i-1}$

- $$\sum_{x \in S} c_x \leq \sum_{i=1}^{k} (u_{i-1} - u_i) \cdot \frac{1}{u_{i-1}} \quad \text{by (4)}$$

$$= \sum_{i=1}^{k} \sum_{j=u_i+1}^{u_{i-1}} \frac{1}{u_{i-1}}$$

$$\leq \sum_{i=1}^{k} \sum_{j=u_i+1}^{u_{i-1}} \frac{1}{j} \quad \text{since } j \leq u_{i-1}$$

$$= \sum_{i=1}^{k} \left(\sum_{j=1}^{u_{i-1}} \frac{1}{j} - \sum_{j=1}^{u_i} \frac{1}{j} \right)$$

$$= \sum_{i=1}^{k} \left(H(u_{i-1}) - H(u_i) \right)$$

$$= H(u_0) - H(u_k) \quad \text{(Telescopic sum)}$$

$$= H(u_0) - H(0)$$

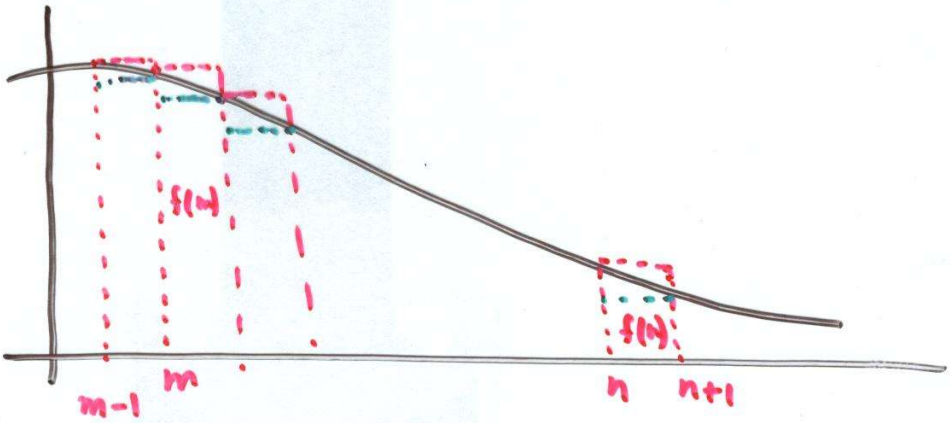
$$= H(u_0)$$

$$= H(|S|).$$

Harmonic Series. We want to estimate $\sum_{k=1}^{n} \frac{1}{k}$.

When $f(x)$ is a monotonically decreasing function

$$\int_m^{n+1} f(x)\,dx \le \sum_{k=m}^{n} f(k) \le \int_{m-1}^{n} f(x)\,dx.$$



Take $f(x) = \frac{1}{x}$.

Then, $\sum_{k=1}^{n} \frac{1}{k} \ge \int_1^{n+1} \frac{dx}{x} = \ln(n+1)$.

Also, $\sum_{k=2}^{n} \frac{1}{k} \le \int_1^{n} \frac{dx}{x} = \ln n$.

So, $\sum_{k=1}^{n} \frac{1}{k} \le \ln n + 1$.

Theorem. Greedy-set-cover is a polynomial time $(\ln|X|+1)$-approximation algorithm.

Proof. Combine the previous results.

In some cases $\max\{|S| : S \in F\}$ is a small constat. In those cases, Approx-set-Cov returns a good approximation. For example, one may apply this algorithm to compute an approximate vertex cover when the graph has maximum degree 3. Then, the subsets have size 3. The greedy-set-cover can return an approximate vertex-cover with $H(3) = 11/6$ approximation ratio.

This is better than what Approx Vertex-Cover will return on such graphs.