CMPSC 465
Data Structures and Algorithms
Spring 2022

Instructor: Chunhao Wang

Greedy algorithms

Greedy algorithms

Set Cover (Textbook Section 5.4)

The set cover problem

Problem (Set Cover)

Input:

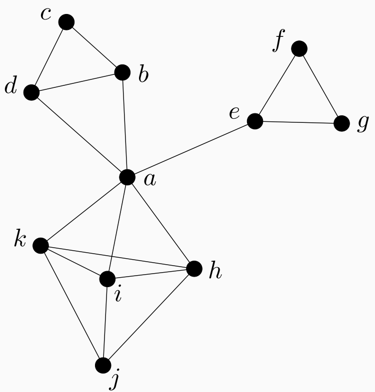
- *a set* B
- *subsets* $S_1, \ldots, S_m \subseteq B$

Output: *a collection of subsets* $S_{i_1}, \ldots, S_{i_k}$ *s.t.* $\bigcup_{j=1}^{k} S_{i_j} = B$

Goal: *minimize the number of selected subsets*

Set cover: example

Example: Each post office can serve 30 miles. Where to build post offices in centre county?



$$B = \{a, b, \dots, k\}$$

$$S_a = \{a, b, d, e, h, i, k\}$$

$$S_b = \{b, c, a, d\}$$

$$\vdots$$

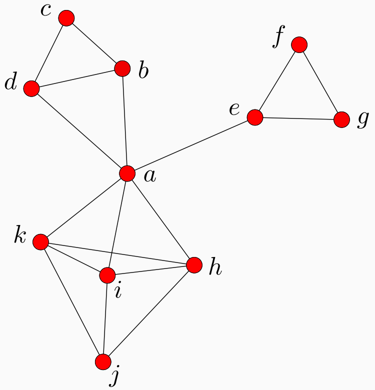
$$S_k = \{k, a, h, i, j\}$$

S_x: the towns within 30 miles of x

Draw an edge if two towns are within 30 miles

Set cover: greedy heuristic

Greedy heuristic: choose the next subset with the most number of uncovered items, until B gets covered



$S_a = \{a, b, d, e, h, i, k\}$

$S_f = \{f, g, e\}$

$S_c = \{c, b, d\}$

$S_j = \{i, k, j, h\}$

Is this optimal?

Optimal solution: S_b, S_e, S_i

Greedy solution is not too bad

Although the greedy solution is not optimal, but it's not off by much

Theorem

Assume $|B| = n$ *and the optimal solution uses* k *subsets. Then the greedy algorithm uses at most* $k \ln(n)$ *subsets*

$\ln(n)$: *approximation ratio*

More about **approximation algorithms**: CSE 565

Proof: Let n_t be the number of elements not covered by the greedy algorithm after t iterations. These remaining n_t elements are covered by the optimal k subsets. So some subsets has $\geq \frac{n_t}{k}$ of these uncovered elements, and the greedy algorithm will pick a set of size at least $\frac{n_t}{k}$.
So, $n_{t+1} \leq n_t - \frac{n_t}{k} = n_t \left(1 - \frac{1}{k}\right)$

Repeatedly applying this:

$$n_t \leq n_{t-1}\left(1 - \frac{1}{k}\right) \leq n_{t-2}\left(1 - \frac{1}{k}\right)^2 \leq \cdots \leq n_0\left(1 - \frac{1}{k}\right)^t = n\left(1 - \frac{1}{k}\right)^t$$

Using the fact: $1 - x \leq e^{-x}$ (equality when $x = 0$)

$$n_t \leq n\left(1 - \frac{1}{k}\right)^t \leq ne^{-t/k}$$

Greedy algorithm terminates when $n_t < 1$. Let's find out what t makes $n_t < 1$

Since $n_t < ne^{-t/k}$, it suffices to make $ne^{-t/k} \leq 1$

Solving $ne^{-t/k} \leq 1$

$\iff e^{-t/k} \leq \frac{1}{n} \iff -\frac{t}{k} \leq \ln(\frac{1}{n}) \iff t \geq -k\ln(\frac{1}{n}) = k\ln(n)$

At $t = k\ln(n)$, $n_t < 1$. Everything is covered $\square$

Proof of the fact $1 - x \leq e^{-x}$ (equality when $x = 0$):

Consider $f(x) = e^{-x} - (1 - x) \geq 0$

$f'(x) = -e^{-x} + 1$. Critical point at $x = 0$, achieving minimum $\square$