Algorithm Analysis and Design (CS1.301)

Monsoon 2021, IIIT Hyderabad
15 September, Wednesday (Lecture 8)

Greedy Algorithms (contd.)

Set Cover Problem

Given a set B and sets $S_1, S_2, \ldots, S_m \subseteq B$, we need to find a selection of the S_i whose union is B. This problem is NP-complete.

The Greedy Algorithm

The greedy algorithm is to continuously pick the S_i with the largest number of uncovered elements.

For example, let

$$B = \{a, d, e, h, i, l, n, o, r, s, t, u\}$$

and let the S_i be

$$\{\text{arid}, \text{dash}, \text{drain}, \text{heard}, \text{lost}, \text{shun}, \text{nose}, \text{slate}, \text{snare}, \text{thread}, \text{lid}, \text{roast}\}.$$

The greedy algorithm then gives us the cover $\{\text{thread}, \text{lost}, \text{drain}, \text{shun}\}$, which can be proved to be a minimum cover.

However, this algorithm does not always lead to an optimum solution. For instance, if $B = \{1, 2, 3, 4, 5, 6\}$, and the S_i are $\{1, 2, 3, 4\}, \{1, 3, 5\}, \{2, 4, 6\}$, then the greedy algorithm returns all three, while only two are enough.

Analysis

If k is the size of the optimal cover, the greedy algorithm uses at most $k \ln n$ sets (where $|B| = n$).

To prove this, let n_t be the number of elements not yet covered after t iterations ($n_0 = n$).

Since these elements are covered by the k sets, there must be some set with at least $\frac{n_t}{k}$ of them. Then

$$n_{t+1} \le n_t - \frac{n_t}{k} = n_t \left(1 - \frac{1}{k}\right),$$

which implies that

$$n_t \le n_0 \left(1 - \frac{1}{k}\right)^t .$$

We know that $1 - x < e^{-x}$ iff $x \neq 0$. Therefore,

$$n_t < n_0 \left(e^{-\frac{1}{k}}\right)^t = ne^{-\frac{t}{k}} .$$

Now $n_t < 1 = ne^{-\ln n}$ at $t = k \ln n$. This completes the proof.