CS 170 Section 4
Greedy Algorithms I

Owen Jow | owenjow@berkeley.edu

Agenda

- Greedy algorithms
- Set cover

Greedy Algorithms

Greedy Algorithms

- A **greedy algorithm** repeatedly selects the best-looking option according to some metric, building up a solution piece by piece.
    - At each step, it chooses the option that seems the most immediately beneficial.
- Often, greedy strategies allow us to find good – but not necessarily optimal – solutions in linear time.

Example	**“Greedy” strategy**
Kruskal’s algorithm for finding an MST	Pick the next lightest edge that doesn’t create a cycle
Huffman encoding [for constructing a coding tree]	Join the two nodes with the lowest frequencies

Set Cover

Set Cover

- We have a collection of elements B, along with S_1, …, S_m which are subsets of B. We would like to determine a minimum-size collection of sets whose union is B. (In other words, we would like to find the minimum number of sets – and the sets themselves – necessary to cover all of the elements in B.)

The natural solution follows a greedy approach: *at every step, pick the set that covers the most uncovered elements of B*.

If the optimal cover consists of k sets, this greedy algorithm will use at most kln(n) sets!