Greedy algorithms

- **Greedy algorithm** paradigm
  - build up a solution incrementally
  - at each step, make the “greedy” choice
- **Example**: in undirected graph $G = (V,E)$, a vertex cover is a subset of $V$ that touches every edge
  - a hard problem: find the smallest vertex cover

Dijkstra’s algorithm

- given
  - directed graph $G = (V,E)$ with non-negative edge weights
  - starting vertex $s \in V$
- find shortest paths from $s$ to all nodes $v$
  - note: unweighted case solved by BFS

Dijkstra’s algorithm

- shortest paths exhibit “optimal substructure” property
  - optimal solution contains within it optimal solutions to subproblems
  - a shortest path from $x$ to $y$ via $z$ contains a shortest path from $x$ to $z$
- shortest paths from $s$ form a tree rooted at $s$
- **Main idea**:
  - maintain set $S \subseteq V$ with correct distances
  - add nbr $u$ with smallest “distance estimate”

**Lemma**: can be implemented to run in $O(m)$ time plus $n$ EXTRACT-MIN and $m$ DECREASE-KEY calls.

**Proof**?
Dijkstra’s algorithm

Lemma: can be implemented to run in \( O(m) \) time plus \( n \) \text{EXTRACT-MIN} and \( m \) \text{DECREASE-KEY} calls.

Proof: each vertex added to \( H \) once, adj. list scanned once, \( O(1) \) work apart from \text{min-heap} calls.

Dijkstra’s algorithm

Lemma: invariant of algorithm: for all \( v \in S \) it \( v\.dist = \text{distance}(s, v) \).

Proof: induction on size of \( S \)
- base case: \( S = \emptyset \), trivially true
- case \(|S| = k\):

Dijkstra’s example from CLRS

Dijkstra’s algorithm

Theorem (Dijkstra): there is an \( O(n \log n + m) \) time algorithm that is given
- a directed graph with nonnegative weights
- a starting vertex \( s \)
and finds
- distances from \( s \) to every other vertex
- (and produces a shortest path tree from \( s \))

\( \quad \text{later: Fibonacci heaps: } O(n \log n + m) \) time

We proved: