Algorithms: Design and Analysis, Part II

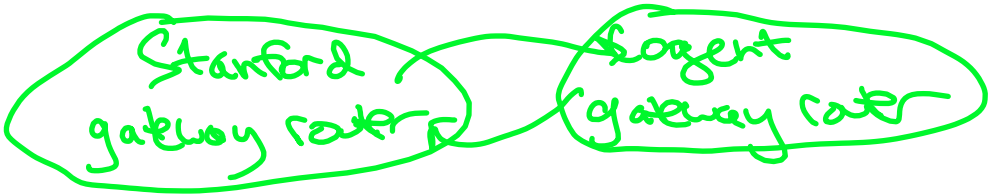
Introduction

Motivating Application: Distributed Shortest-Path Routing

Graphs and the Internet

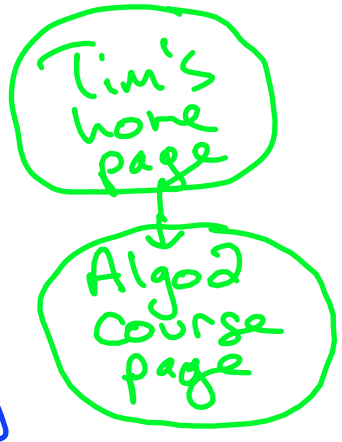
Claim: the Internet is a graph.

[vertices = end hosts + routers, directed edges = direct physical or wireless connections]



Other graphs related to the Internet:

- Web graph [vertices = Web pages, edges = hyperlinks]
- Social networks [vertices = people, edges = friend/follow relationships]



Internet Routing

Suppose: Stanford gateway router needs to send data to the Cornell gateway router (over multiple hops).

Question: which Stanford ⇝ Cornell route to use?

Obvious idea: how about the shortest? (e.g., fewest # of hops)

⇒ need a shortest-path algorithm

Recall from Part I: Dijkstra's algorithm does this (with nonnegative edge lengths)

Issue: Stanford gateway rater would need to know entire Internet!

⇒ need a shortest-path algorithm that uses only local computation

Solution: the Bellman-Ford algorithm (bonus: also handles negative edge costs)