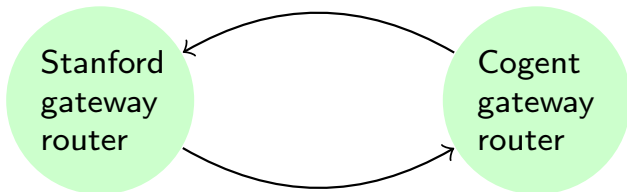
Algorithms: Design and Analysis, Part II

Introduction

Motivating Application: Distributed Shortest-Path Routing

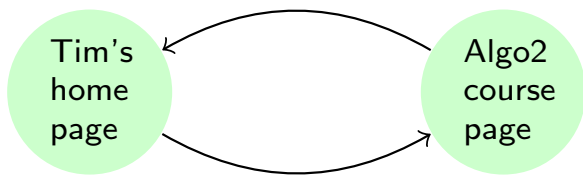
Graphs and the Internet

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 router 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).