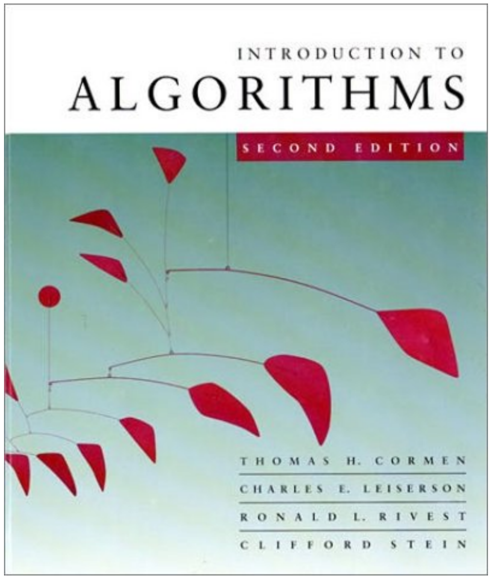
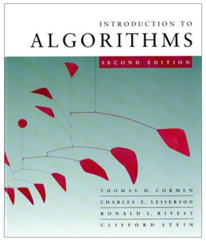
Introduction to Algorithms

6.006



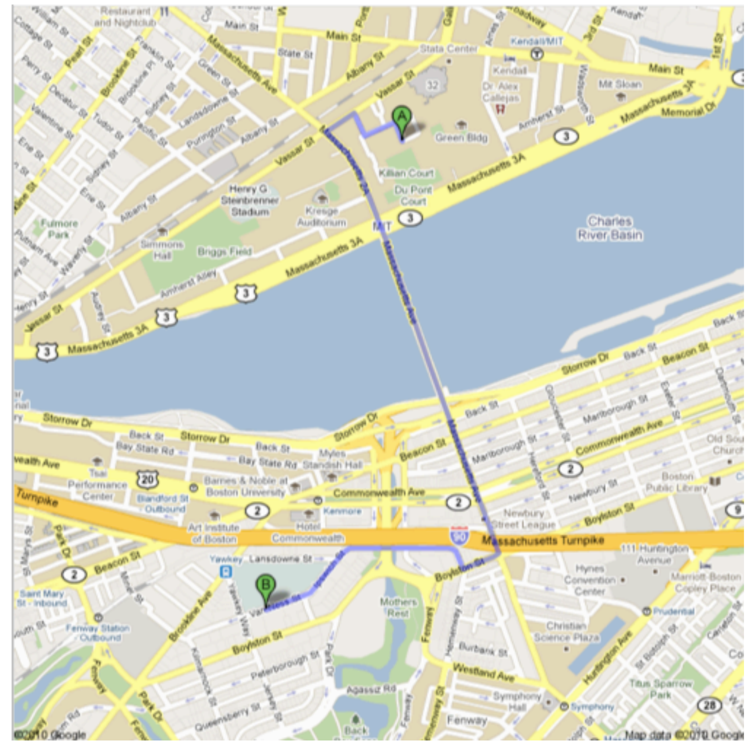
Lecture 17

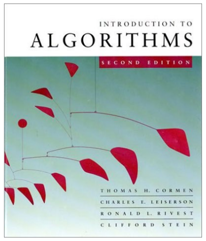
Prof. Piotr Indyk



Menu

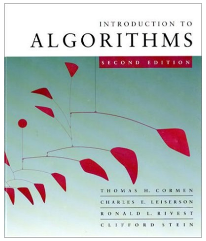
- Last two weeks
  - Bellman-Ford
    - O(VE) time
    - general weights
  - Dijkstra
    - O((V+E)logV) time
    - non-negative weights
- Today: applications
  - Obstacle course for robots
  - Scheduling with constraints





Obstacle course for robots

- Obstacles: disjoint triangles $T_1 \ldots T_n$
- Robot: a point at position A
- Goal: the shortest route from A to B

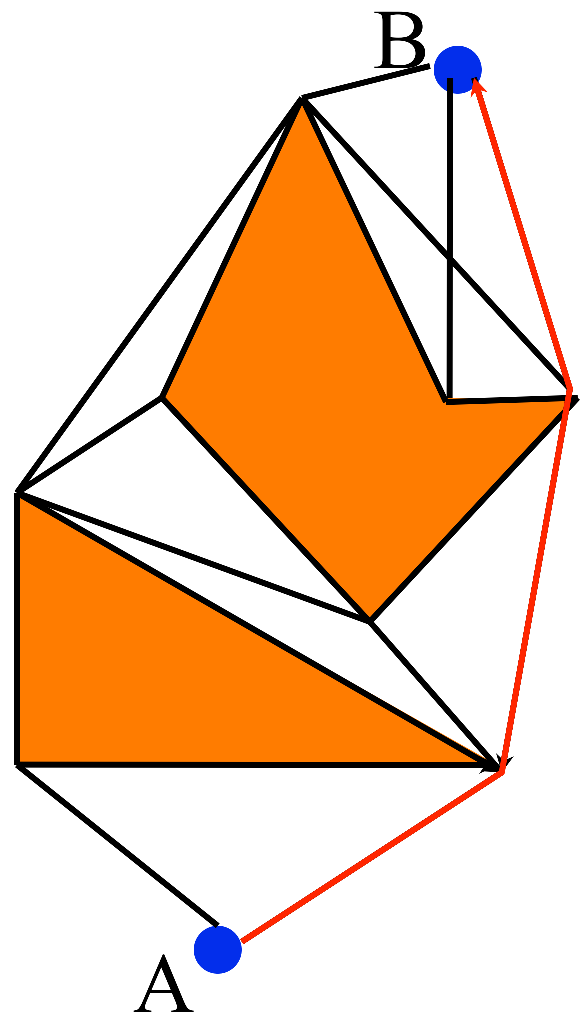


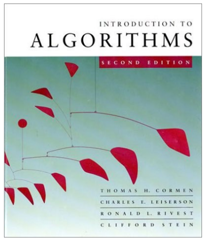
Path planning algorithm

- Let V be the set consisting of triangle vertices, A and B
  - Note that V=O(n)
- Observation: the shortest path consists of line segments between points in V
- Approach:
  - For each pair u,v in V such that the segment u-v is "free", create an edge u-v

    (weight = segment length).

    This is called **visibility graph** G
  - Compute the shortest path from A to B in G





Computing visibility graph

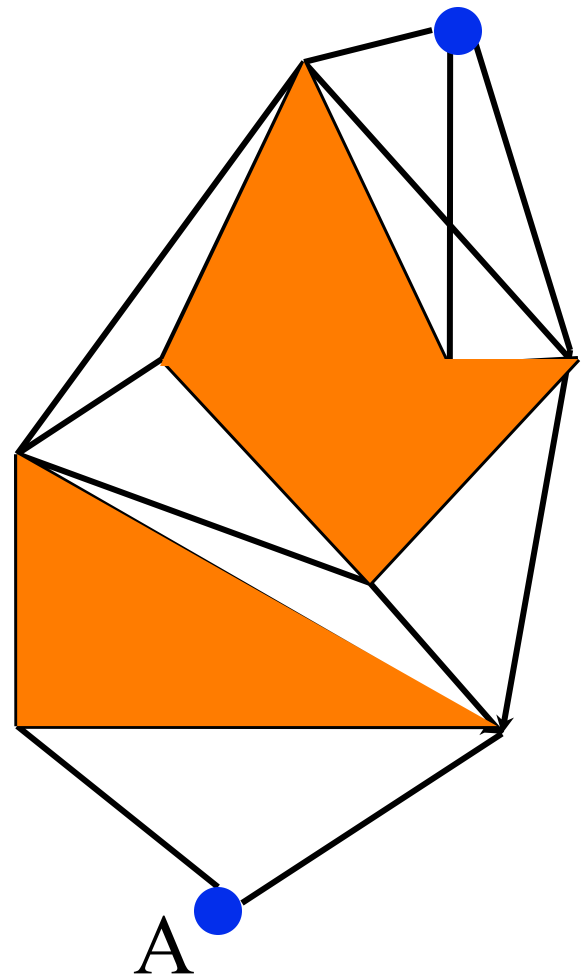
- For each segment u-v, check whether there is any triangle T_i such that one of its sides (say, s) intersects u-v
  - The test whether s intersects u-v can be done in O(1) time

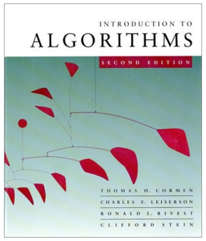
  [CLRS 33.1, or lecture 24]
- Time: $O(V^3)$

  [Lozano-Perez'79]
- Total time for path planning?

  $O(V^3)$
- Best known: O(V log V)

  [Hershberger-Suri'97]





Solving a system of difference constraints

Difference constraints: a system of linear inequalities of the form $x_j - x_i \le w_{ij}$

Example:

$$x_1 - x_2 \le 3$$
$$x_2 - x_3 \le -2$$
$$x_1 - x_3 \le 2$$

Solution:

$$x_1 = 3$$
$$x_2 = 0$$
$$x_3 = 2$$

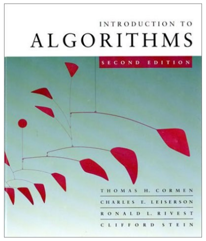
Application: parallel task scheduling with precedence constraints:

- x_i : the starting time of the job i
- If a job i needs to be finished before job j starts:

$$x_j \ge x_i + \text{duration}(i)$$

- The time from start to finish should be at most t:

$$x_j + \text{duration}(j) - x_i \le t \quad \text{for all } i,j$$

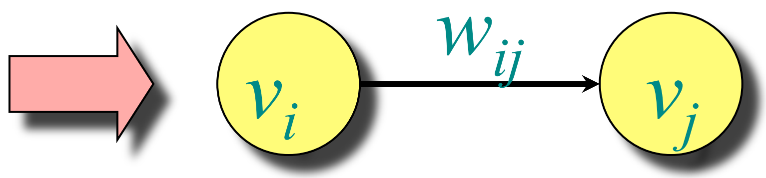


Solving difference constraints via shortest paths

Constraint graph:

- A vertex v_i for each variable x_i
- An edge for each constraint:

$$x_j - x_i \le w_{ij}$$



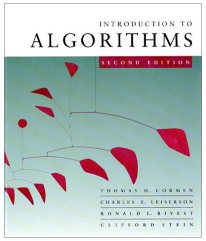
Unsatisfiable constraints

Theorem. If the constraint graph contains a negative-weight cycle, then the system of differences is unsatisfiable.

Proof. Suppose that the negative-weight cycle is $v_1 \to v_2 \to \cdots \to v_k \to v_1$. Then, we have

$$
\begin{aligned}
x_2 - x_1 &\le w_{12} \\
x_3 - x_2 &\le w_{23} \\
&\vdots \\
x_k - x_{k-1} &\le w_{k-1,\,k} \\
x_1 - x_k &\le w_{k1} \\
\hline
0 &\le \text{weight of cycle} < 0
\end{aligned}
$$

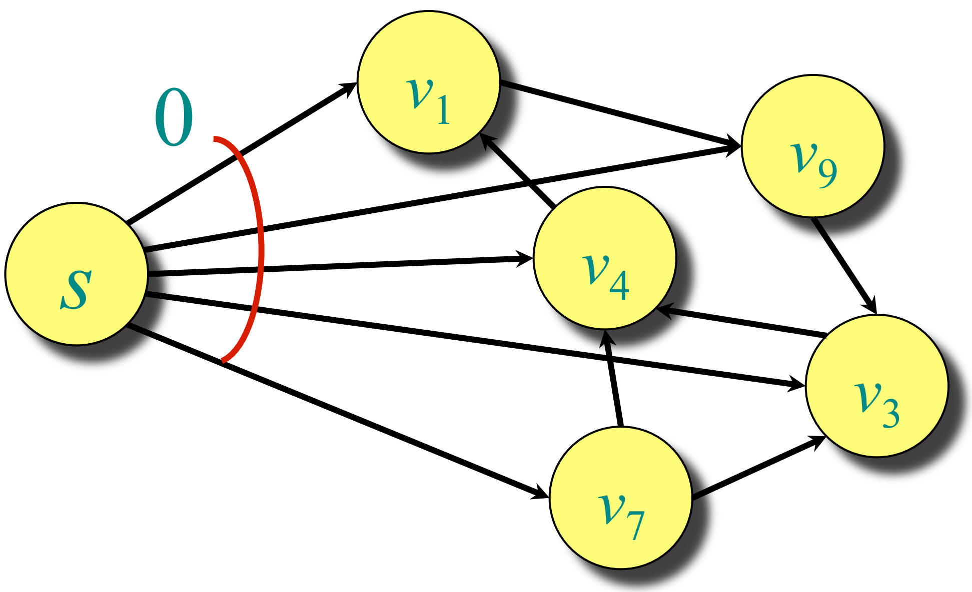
Therefore, no values for the x_i can satisfy the constraints. □



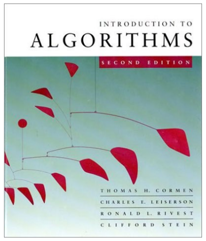
Satisfying the constraints

Theorem. Suppose no negative-weight cycle exists in the constraint graph. Then, the constraints are satisfiable.

Proof. Add a new vertex s to V with a 0-weight edge to each vertex $v_i \in V$.



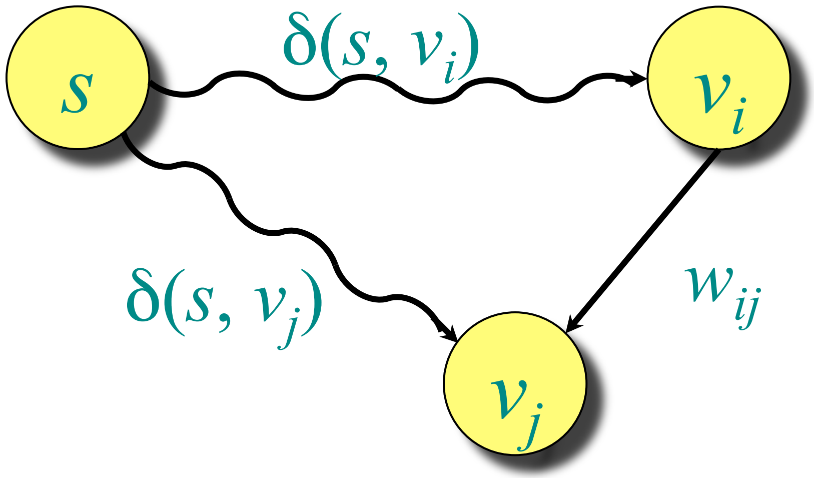
Note:
No negative-weight cycles introduced ⇒ shortest paths exist.



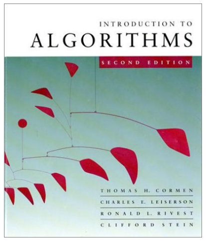
Proof (continued)

Claim: The assignment $x_i = \delta(s, v_i)$ solves the constraints.

Consider any constraint $x_j - x_i \leq w_{ij}$, and consider the shortest paths from s to v_j and v_i:



The triangle inequality gives us $\delta(s,v_j) \leq \delta(s, v_i) + w_{ij}$. Since $x_i = \delta(s, v_i)$ and $x_j = \delta(s, v_j)$, the constraint $x_j - x_i \leq w_{ij}$ is satisfied. □



Bellman-Ford

Corollary. The Bellman-Ford algorithm can solve a system of m difference constraints on n variables in $O(mn)$ time.

Note: Bellman-Ford also minimizes

$$\max_i\{x_i\} - \min_i\{x_i\}$$

(exercise)