All-Pairs Shortest Paths (APSP)

Optimal Substructure

Algorithms: Design and Analysis, Part II

Motivation

Floyd-Warshall algorithm: $O(n^3)$ algorithm for APSP.

- Works even with graphs with negative edge lengths.

Thus: (1) At least as good as n Bellman-Fords, better in dense graphs.

(2) In graphs with nonnegative edge costs, competitive with n Dijkstra's in dense graphs.

Important special case: Transitive closure of a binary (i.e., all-pairs reachability) relation.

Open question: Solve APSP significantly faster than $O(n^3)$ in dense graphs?

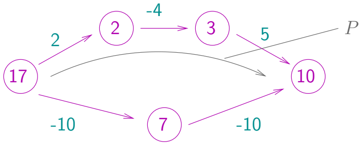
Optimal Substructure

Recall: Can be tricky to define ordering on subproblems in graph problems.

Key idea: Order the vertices $V = \{1, 2, \ldots, n\}$ arbitrarily. Let $V^{(k)} = \{1, 2, \ldots, k\}$.

Lemma: Suppose G has no negative cycle. Fix source $i \in V$, destination $j \in V$, and $k \in \{1, 2, \ldots, n\}$. Let $P =$ shortest (cycle-free) i-j path with $\underline{\text{all internal nodes in } V^{(k)}}$.

Example: $[i = 17, j = 10, k = 5]$



Optimal Substructure (con'd)

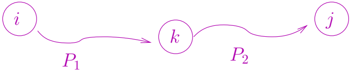
Optimal substructure lemma: Suppose G has no negative cost cycle. Let P be a shortest (cycle-free) i-j path with all internal nodes in $V^{(k)}$. Then:

Case 1: If k not internal to P, then P is a shortest (cycle-free) i-j path with all internal vertices in $V^{(k-1)}$.

Case 2: If k is internal to P, then:

P_1 = shortest (cycle-free) i-k path with all internal nodes in $V^{(k-1)}$ and

P_2 = shortest (cycle-free) k-j path with all internal nodes in $V^{(k-1)}$



Proof: Similar to Bellman-Ford opt substructure (you check!)