Algorithms: Design and Analysis, Part II

All-Pairs Shortest Paths (APSP)

Optimal Substructure

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.

i.e., all-pairs reachability

Important special case: transitive closure of a binary relation.

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

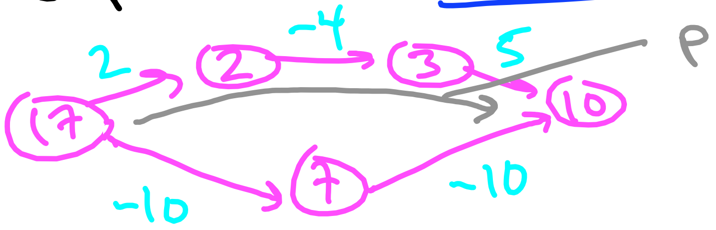
Optimal Substructure

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

Key idea: order the vertices $V = \{1,2,3,\ldots,n\}$ 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 $k \in \{1,2,\ldots,k\}$. Let P = shortest (cycle-free) i-j path with all internal nodes in $V^{(k)}$.

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

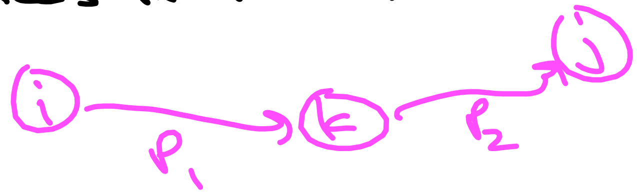


Optimal Substructure (con'd)

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!).