Algorithm Analysis and Design (CS1.301)

Monsoon 2021, IIIT Hyderabad
29 September, Wednesday (Lecture 12)

Dynamic Programming (contd.)

Shortest Reliable Paths

Given a graph G with lengths on the edges, two special nodes s, t and an integer k, we want the shortest path from s to t that uses at most k edges.

For all $i \leq k$, let $\text{dist}(v, i)$ be the length of the shortest path from s to v that uses i edges. Then, we know that

$$\text{dist}(v, i) = \min_{(u,v) \in E} \{\text{dist}(u, i-1) + l(u, v)\}.$$

Thus we can maintain a table of the values of dist to find $\text{dist}(t, k)$.

All Pairs Shortest Path

Using Dijkstra's single-source shortest path algorithm gives us a time of $O(|V|^2|E|)$. The Floyd-Warshall algorithm (following) gives us $O(|V|^3)$.

Let $\text{dist}(i, j, k)$ be the length of the shortest path from i to j in which only nodes $\{1, 2, \dots, k\}$ can be used as intermediates. Then,

$$\text{dist}(i, j, k) = \min\{\text{dist}(i, k, k-1) + \text{dist}(k, j, k-1), \text{dist}(i, j, k-1)\}.$$

Independent Set in Trees

An independent set in a graph $G = (V, E)$ is a set of nodes with no edges between any pair of them. We need to find the maximum independent set in a given tree.

Let $I(u)$ be the size of the largest independent set in the subtree hanging from u. To compute $I(u)$, consider two cases: any independent set either includes u or doesn't. If it does, it cannot include its children; this gives us

$$I(u) = \max \left\{ 1 + \sum_{w \in \text{grandchildren of } u} I(w), \sum_{w \in \text{children of } u} I(w) \right\}.$$