28.3 Longest Path

In General

Consider the problem of finding the longest path from a start vertex to every other vertex. The path must be simple (contain no cycles).

It turns out that best known algorithm is exponential (impractically inefficient).

Negating all the edge weights and finding the shortest path leaves us in a tricky situation because then we could have negative cycles and we could go around and around them indefinitely.

Longest Paths on DAGs

But what if we are dealing with DAGs? In that case, we have no cycles so we can do as suggested above:

1. Form a new copy of the graph, called G', with all edge weights negated (signs flipped).
2. Run DAG shortest paths on G' yielding result X
3. Flip the signs of all values in X.distTo. X.edgeTo is already correct.

Alternatively, we could modify the DAG shortest path algorithm from the previous section to choose the larger distTo when relaxing an edge. While this would make more sense in practice, the benefit of thinking of the first approach is that wree a able to use an existing algorithm as a "black box" to solve a new problem. We'll learn more about this kind of problem solving in the next section: Reductions.

Previous
28.2 Shortest Paths on DAGs

Next

28.4 Reductions and Decomposition

Last updated 1 year ago