Applied Algorithms	February 14, 2012
Bellman-Ford algorithm	
Professor: Antonina Kolokolova	Scribe: Abdullah Al Mamun

1 Bellman-Ford algorithm

Finding shortest path in a graph with only positive weights can be done by Dijkstra's algorithm. The problem becomes a lot harder for the graphs that contain negative weights: in the worst case, a path from s to t may go through a negative cycle, so the shortest path would have weight of negative infinity. Thus, a different algorithm is needed, both to detect the presence of negative cycles in the graph (reachable from s, at least), and to find a shortest path when there are negative weights, although no negative cycles are reachable. The (standard) Bellman-Ford algorithm achieves both goals. There is a different version of this algorithm in Kleinberg-Tardos, which assumes that there are no negative cycles; their version does fit into the dynamic programming paradigm better.

"Standard" Version

Problem: s-t shortest path with negative weights
Initialize:
s.distance = 0;
$\forall\ v \neq s\ v.distance = \infty$
for $i = 1$ to $n - 1$ do
for every edge (u, v) do
if $v.distance > u.distance + weight(u.v)$
then $v.distance \leftarrow u.distance + weight(u, v)$

For every edge (u, v) do
if $v.distance > u.distance + weight(u.v)$
output "negative cycle"

Why does this algorithm works? Note that the longest possible path from s to t is of length $n - 1$. After 1 step, all edges connected by an edge to the source will get non-infinite values. Moreover, all paths from s of length 1 will get their correct shortest path values after the first step. In general, if the best path from s to t has i edges, after i steps it will be computed correctly. Since without negative cycles the best path from s to t will have at most $n - 1$ edges, it will be computed correctly after $n - 1$ steps.

KT Version

1) A(i,v): best path from v to t using at most i edges.
Answer: A(n-1, s)
2) A(i, v) = min{A(i-1, v), min{A(i-1, w) + weight(u, v)}}

Here, the assumption is that the graph contains no negative cycle.