Dijkstra single-source shortest paths. Source = L.
Priority queue = [L 0, H +inf, F +inf, T +inf, J +inf, G +inf, M +inf, K +inf]
extractMin() -> L.
Priority queue = [H +inf, F +inf, T +inf, J +inf, G +inf, M +inf, K +inf]
Relaxing the edges from L. Considering edge (L, M), leading to M.
Priority queue = [H + inf, F + inf, T + inf, J + inf, G + inf, M + inf, K + inf]
$0 + 1 < +\infty$. Edge (L, M) relaxed.
Finished with L.
Priority queue = [M 1, H +inf, F +inf, T +inf, J +inf, G +inf, K +inf]
extractMin() -> M.
Priority queue = [H +inf, F +inf, T +inf, J +inf, G +inf, K +inf]
Relaxing the edges from M. Considering edge (M, T), leading to T.
Priority queue = [H +inf, F +inf, T +inf, J +inf, G +inf, K +inf]
$1 + 8 < +\text{inf. }$ Edge $(M, T)$ relaxed.
Relaxing the edges from M. Considering edge (M, H), leading to H.
Priority queue = [T 9, H +inf, F +inf, J +inf, G +inf, K +inf]
$1 + 6 < +\infty$. Edge (M, H) relaxed.
Finished with M.
Priority queue = [H 7, T 9, F +inf, J +inf, G +inf, K +inf]
extractMin() -> H.
Priority queue = [T 9, F +inf, J +inf, G +inf, K +inf]
Relaxing the edges from H. Considering edge (H, K), leading to K.

Priority queue = [T 9, F +inf, J +inf, G +inf, K +inf]
$7 + 7 < +\infty$. Edge $(H, K)$ relaxed.
Relaxing the edges from H. Considering edge (H, J), leading to J.
Priority queue = [T 9, K 14, F +inf, J +inf, G +inf]
\[7 + 2 < +\infty.\] Edge \((H, J)\) relaxed.
Relaxing the edges from H. Considering edge (H, T), leading to T.
Priority queue = [T 9, J 9, K 14, F +inf, G +inf]
7 + 32 is not < 9. Edge (H, T) not relaxed.
Finished with H.
Priority queue = [T 9, J 9, K 14, F +inf, G +inf]
extractMin() -> T.
Priority queue = [J 9, K 14, F +inf, G +inf]
Relaxing the edges from T. Considering edge (T, G), leading to G.
Priority queue = [J 9, K 14, F +inf, G +inf]
9 + 10 < +inf. Edge (T, G) relaxed.
Relaxing the edges from T. Considering edge (T, J), leading to J.
Priority queue = [J 9, K 14, G 19, F +inf]
9 + 4 is not < 9. Edge (T, J) not relaxed.
Finished with T.
Priority queue = [J 9, K 14, G 19, F +inf]
extractMin() \rightarrow J.
Priority queue = [K 14, G 19, F +inf]
Relaxing the edges from J. Considering edge (J, L), leading to L. Priority queue = [K 14, G 19, F +inf]
9 + 5 is not < 0. Edge (J, L) not relaxed.
Relaxing the edges from J. Considering edge (J, K), leading to K. Priority queue = [K 14, G 19, F +inf]
9 + 23 is not < 14. Edge (J, K) not relaxed.
Finished with J.
Priority queue = [K 14, G 19, F +inf]
extractMin() -> K.
Priority queue = [G 19, F +inf]
Finished with K.
Priority queue = [G 19, F +inf]
extractMin() -> G.
Priority queue = [F +inf]
Finished with G.
Priority queue = [F + inf]
extractMin() -> F.
Priority queue = []
Relaxing the edges from F. Considering edge (F, H), leading to H.

Priority queue = []
+inf + 3 is not < 7. Edge (F, H) not relaxed.
Relaxing the edges from F. Considering edge (F, L), leading to L.

Priority queue = []
+inf + 3 is not < 0. Edge (F, L) not relaxed.
Relaxing the edges from F. Considering edge (F, G), leading to G. 

Priority queue = []
+inf + 5 is not < 19. Edge (F, G) not relaxed.
Finished with F.
Priority queue = []
Dijkstra single-source shortest paths
Generated by $Id: dijkstra.py 99 2010-11-18 10:48:22Z fms27 $
(c) 2006-2010 Frank Stajano
University of Cambridge Computer Laboratory