1 Conditions for Shortest Path Algorithms

Suppose that we want to find the shortest path between two nodes in the following graph. Which algorithms can we use?

A. BFS
B. Dijkstra
C. Bellman-Ford
D. All of the above

We have a directed graph with nonnegative edge weights. Can we use Dijkstra to find shortest paths?

A. Yes
B. No

We have a graph with negative edge weights. Can we use Dijkstra to find shortest paths?

A. Yes
B. No

We have a directed graph with positive edge weights. Can we use Dijkstra to find shortest paths?

A. Yes
B. No

2 Dijkstra Forensics

Suppose we use Dijkstra on some graph with nodes A, B, C, D, E, F that has nonnegative (≥ 0) edge weights, starting from the node A (in the middle of the algorithm) our computer crashes. We look through the memory dump and we see that the size of a bucket is 8. Follow what the crash happened:

\[d[A] = 0.0, d[B] = 0.0, d[C] = 4.0, d[D] = 2.0, d[E] = 20.0\]

Additionally from the memory dump we see that the current node when the crash happened was node C.

What is the maximum possible length of the shortest path from node A to node B?

A. Correct
B. 4
C. 8
D. 15

What is the maximum possible length of the shortest path from node A to node D?

A. Correct
B. 4
C. 8
D. 15

What is the maximum possible length of the shortest path from node A to node E?

A. Correct
B. 4
C. 8
D. 15

What is the maximum possible length of the shortest path from node A to node F?

A. Correct
B. 4
C. 8
D. 15

If we run the Dijkstra algorithm on the graph of U.S. streets/roads/highways/etc., starting from the Tresidder Union Times Square in New York, can we use Dijkstra?

A. Yes
B. No

3 Runtime

Suppose that we implement Dijkstra with a red-black tree. What is the asymptotically smallest upper bound on the runtime in terms of \(n\) (the number of nodes) and \(m\) (the number of edges).

A. \(O(\log n)\)
B. \(O(m \log n)\)
C. \(O(n \log n)\)
D. \(O(n m)\)

What if we implement Dijkstra with a Fibonacci heap? What is the asymptotically smallest upper bound on the runtime in terms of \(n\) (the number of nodes) and \(m\) (the number of edges).

A. \(O(\log n)\)
B. \(O(m \log n)\)
C. \(O(n \log n)\)
D. \(O(n m)\)

What is the asymptotically smallest upper bound on the runtime of the above code assuming that both the insert and remove operations on \(H\) take \(O(\log |H|)\) time? Assume that \(n \leq m\) (in particular \(n \leq \log m\)).

A. \(O(\log n)\)
B. \(O(m)\)
C. \(O(n \log n)\)
D. \(O(n m)\)