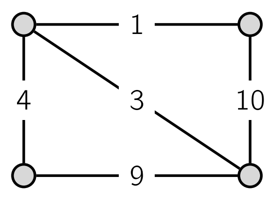
Dijkstra

Reset Progress Reveal Solutions

1 Conditions for Shortest Path Algorithms

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



○ BFS
○ Dijkstra
○ Bellman-Ford
○ All of the above
○ BFS and Dijkstra
● Dijkstra and Bellman-Ford

Correct

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

○ Yes
● No

Correct

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

● Yes
○ No

Correct

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

● Yes
○ No

Correct

2 Dijkstra Forensics

Suppose we run 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 see that the state of d looked as follows when the crash happened:

$$d[A] = 0, d[B] = 5, d[C] = 4, d[D] = 15, d[E] = 2, d[F] = 20.$$

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

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

4

Correct

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

5

Correct

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

4

Correct

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

15

Correct

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

2

Correct

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

2

Correct

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

4

Correct

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

20

Correct

If we run the Dijkstra algorithm on the graph of U.S. streets/roads/highways/etc., starting from the Stanford Oval, which of the following locations will become the current node first?

○ Times Square in New York
○ The Hollywood Sign
● Tresidder Union
○ The ordering might differ in each run of Dijkstra.

Correct

3 Runtime

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

○ $O(n \log n + m)$
● $O((n + m) \log n)$
○ $O(n + m)$

Correct

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

● $O(n \log n + m)$
○ $O((n + m) \log n)$
○ $O(n + m)$

Correct

Suppose that we have a heap data structure that does not support updating the keys (many standard implementations of heaps in various programming languages do not support the update key operation).

Our data structure keeps a collection of items, each of form (key, object), where keys are numbers, and object can be anything (we will store vertices as our objects). Our data structure supports two operations:

- Insert a new (key, object) into the collection.
- Remove the item with the lowest key currently in the collection and return the key and object for it.

We run a modification of Dijkstra with the following pseudo-code:

$d \leftarrow$ array indexed with vertices and filled with ∞
$H \leftarrow$ empty heap
Insert $(0, \text{starting node})$ into H
while *H is not empty* **do**
  Remove (key, vertex) from H with the smallest key.
  if $key < d[vertex]$ **then**
    $d[\text{vertex}] \leftarrow \text{key}$
    for *all neighbors w of vertex* **do**
      Insert $(d[\text{vertex}] + \text{weight}(\text{vertex}, w), w)$ into H.

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 $O(\log(\text{size of } H))$ time? Assume that $n - 1 \leq m \leq n^2$ (in particular $\log m = \Theta(\log n)$).

○ $O(n \log n + m)$
● $O(m \log n)$
○ $O(n + m)$

Correct