9 Algorithms 2 (djw1005)

We are given a directed graph g with edge costs ≥ 0, and we wish to find the distance between two given vertices s and t. Your friend has the idea that we should waste less time exploring irrelevant parts of the graph, and suggests the following procedure:

"Run the standard version of Dijkstra's algorithm `dijkstra(`g`,`s`)` starting at s; and also run a variant `artskjid(`g`,`t`)` that starts at t and finds distances *to* t. Interleave these two by visiting one vertex with `dijkstra`, then one with `artskjid`, then one with `dijkstra`, and so on; terminate when one of them visits a vertex m that the other has already visited. Let $d = m$`.distance` be the distance computed by `dijkstra`, and let $e = m$`.ecnatsid` be the distance computed by `artskjid`; and return $d + e$."

(*a*) Explain how to implement `artskjid(`g`,`t`)` efficiently. What is the worst-case asymptotic running time of `artskjid`? [3 marks]

(*b*) Does your friend's procedure improve on the asymptotic worst-case time of simply running `dijkstra(`g`,`s`)`? Justify your answer. [8 marks]

(*c*) Your friend gives the following argument for correctness: "Since `dijkstra` visits vertices in order of increasing distance from s, and `artskjid` visits in order of increasing distance to t, the point where they meet must be on the shortest path from s to t."

Your friend's procedure can in fact give an incorrect answer. Demonstrate the problem with your friend's reasoning. [9 marks]