

Let k be any positive integer. Consider a graph with $n = 2k + 1$ vertices, numbered 1 through $2k + 1$, with arcs $(j, j - 1)$ for each j from 2 to $2k + 1$ and arcs $(2j + 1, 2j - 1)$ for each j from 1 to k . Thus the graph consists of a path from $2k + 1$ to 1 through consecutively decreasing vertices, plus “shortcuts” between each consecutively decreasing pair of odd vertices, for a total of $3k$ arcs. The arcs have the following costs:

$c(2j + 1, 2j - 1) = 2^{j-1}$, $c(2j + 1, 2j) = 2^j$, $c(2j, 2j - 1) = -2^j$ for each j from 1 to k .

When considering versions of the scanning algorithm, assume that for each vertex v the list of arcs (v, w) from v are ordered in increasing order by w . That is, the list for vertex 7 contains $(7, 5)$ followed by $(7, 6)$, so $(7, 5)$ will be scanned (relaxed) before $(7, 6)$. When considering breadth-first labeling, assume the list of all the arcs (v, w) (through which the algorithm makes passes) is in increasing order by v , with ties broken in increasing order by w . That is, the list is $(2, 1)$, $(3, 1)$, $(3, 2)$, $(4, 3)$,...

(1) What is the length of a shortest path from vertex $2k + 1$ to vertex 1? Is the shortest path unique? If so, how many arcs are on it; if not, how many arcs are on a shortest path of fewest arcs?

(2) Is the graph acyclic? If not, give a cycle. If so, give a topological order.

(3) Suppose we want to run some version of the scanning algorithm starting from source $2k + 1$ to find a shortest path from $2k + 1$ to 1. Which version of the algorithm would you run (breadth-first scanning, shortest-first scanning, or topological scanning) and why?

(4) Draw the graph for $k = 3$. On this graph, how many arc scans (relaxations) will each of the following algorithms do? (A scan counts even if it does not decrease a distance.)

(a) Breadth-first labeling.

(b) Breadth-first scanning (without subtree disassembly).

(c) Breadth-first scanning with subtree disassembly but without distance updates.

(d) Shortest-first scanning if a vertex, once scanned, is never reinserted into the heap L . When deleting from the heap, break a tie in favor of the vertex with lowest number.

(e) Shortest-first scanning, if a vertex is reinserted into L any time it gets a smaller label, even if it has already been scanned. When deleting from the heap, break a tie in favor of the vertex with lowest number.

(f) Topological scanning.

(5) For each of the six algorithms considered in problem (4), give the number of arc scans the algorithm would do on the graph with an arbitrary value of k , as a function of k . For full credit, give an exact answer; for partial credit, give an answer correct to within a constant factor if the answer is polynomial, or correct to within a constant factor in the exponent if the answer is exponential.

