Activity 16

Pseudocode

Distance between JHB and DBN using two different algorithms

Dijkstra's algorithm

function dijkstra(graph, start):

create priority queue pq

create dictionary distances with default value infinity

set distances[start] to 0

push (0, start) into pq

while pq is not empty:

(current\_distance, current\_node) = pop from pq

if current\_distance > distances[current\_node]:

continue

for each (neighbor, weight) in graph[current\_node]:

distance = current\_distance + weight

if distance < distances[neighbor]:

distances[neighbor] = distance

push (distance, neighbor) into pq

return distances

Bellman\_ford

function bellman\_ford(graph, start):

create dictionary distances with default value infinity

set distances[start] to 0

for i from 1 to number of nodes - 1:

for each node in graph:

for each (neighbor, weight) in graph[node]:

if distances[node] + weight < distances[neighbor]:

distances[neighbor] = distances[node] + weight

for each node in graph:

for each (neighbor, weight) in graph[node]:

if distances[node] + weight < distances[neighbor]:

raise error "Graph contains a negative weight cycle"

return distances