Mandatory Assignement 3.

1. See the hand-drawn diagram.
2. See the hand-drawn diagram.
3. PROBLEM

Find which intermediate stations (Ui, Vj) to pass through to achieve the shortest route between stations X and Y.

Station X is located in Zone 1 (Z1)

Stations Ui are located in Zone 2 (Z2)

Stations Vj are located in Zone 3 (Z3)

Station Y is located in Zone 4 (Z4)

All stations in Z1 are directly connected to all stations in Z2, the same applies for Z2 to Z3 and Z3 to Z4.

There are m amount of stations in Z2.

There are n amount of stations in Z3.

The length between X and Ui is ai.

The length between Ui and Vj is bij.

The length between Vj and Y is cj.

Total length = ai + bij + cj.

ALGORITHM

PRECONDITIONS

The amount of stations in Z2 (m) most be determined.

The amount of stations in Z3 (n) most be determined.

The length between each station most be supplied.

POSTCONDITIONS

No supplied data has been modified.

The i and j values that result in the shortest route will be returned in a set.

STEPS IN ALGORITHM

1. Save the length of any possible route for comparison purposes. Save the i and j values that resulted in the length.
2. Loop through all possible values of i while performing step 3 with each loop.
3. Loop through all possible values of j while performing step 4 with each loop.
4. Determine the length that the current i and j values result in, compare with the saved length, if shorter overwrite saved length with new length and saved i and j values with corresponding new i and j values.
5. Done.

STEPS IN THE ALGORITHM – PSUEDOCODE

func intermediateStations(a(m), b(m)(n), c(n)) {

var stations(2) = (0,0)

var savedLength = a(0) + b(0)(0) + c(0)

var tempLength = 0

for (var i = 0; i < m; i++) {

for (var j = 0; j < n; j++) {

tempLength = a(i) + b(i)(j) + c(j)

if (tempLength < savedLength) {

savedLength = tempLength

stations(0) = i

stations(1) = j

}

}

}

Return stations;

1. Done