**PROBLEM**

Given a certain number of stations, and the distance taken to travel between stations with direct access to one another, find the shortest path from station X to station Y, using intermediate stations.

**ALGORITHM**

Calculate all possible paths from station X to station Y (excluding those that backtrack) and find the shortest one.

**PRECONDITIONS**

* There must be direct paths from X to any station in Zone 2 (a\_i), any station in Zone 2 to any station in Zone 3 (b\_ij), and any station in Zone 3 to station Y (c\_j)
* There must be at least one station in Zone 2 as well as Zone 3

**POSTCONDITIONS**

* The program will include a method that returns the shortest path possible from station X to station Y
  + The first index will hold which station in Zone 2 (marked from 0 to m-1) is to be followed in the shortest path
  + The second index will hold which station in Zone 3 (marked from 0 to n-1) is to be followed in the shortest path
* A method will be available that returns the length of the shortest path from station X to station Y

**STEPS IN THE ALGORITHM**

1. Assume the shortest path to be the first one, and hold this value
2. Iterate through the possible paths from station X to all stations in Zone 2
3. Inside this iteration, iterate through the possible paths from all stations in Zone 2 to all stations in Zone 3
4. Add the lengths of these two intermediate paths, as well as the path from any station in Zone 3 to station Y, and hold this value
5. If the value obtained in step 4 is smaller than the value obtained in step 1, swap out the value in step 1 for the one in step 4

**STEPS IN THE ALGORITHM – PSEUDOCODE**

* Start Program [inputs are two one dimensional arrays: a and c, and a two-dimensional array: b]
* Variable L = a\_0 + b\_0\_0 + c\_0
* Array P =
* Loop(do i = 1, while i < length(a), then do i = i + 1):
  + Loop(do j = 1, while j < length(c), then do j = j + 1):
    - If(a\_i + b\_i\_j + c\_j < L):
      * L = a\_i + b\_i\_j + c\_j
      * P\_0 = i
      * P\_1 = j
* Return L