Post – Disaster Management

The main goal of this hackathon statement is to develop a code that efficiently plans routes when a disaster occurs such as cyclone,floods ,etc managing the given constraints like vehicles,road damage and resources.

After a disaster (for e.g.:- recent cyclones), many places got damaged and are short of resources. Your task is to distribute resources efficiently.

Scenario :-

Locations and damage :

Let's say there are 'k' locations A,B,C…. and so on where the resource distribution occurs, and some resource supply centres.
Some(or all) of the locations among A,B,C,….. are sort of the resources and you are provided with 'n' no.of small and large types of vehicles in total. Roads are damaged so small trucks can travel on roads with damage level – 0 or 1 whereas large vehicles can travel on roads only with damage level – 0. Damage at each point is known. Path

Threshold,sources and distribution of resources :

There is a threshold level, below which if the resources are available at a place, it becomes a deficient place (destination). **If the resources available at a place are above threshold level, it can be a source.** In that case the resources can be taken even from those places in addition to the supply centres.

If the **supply centre (or resource centre)** (multiple supply centres) is(are) some of the sources among multiple sources to a destination, the destination receives resources in an extra amount of **X**. If multiple supply centres are acting as sources to a destination, the source for the extra X is one among them as decided by you.

If all the sources to a destination are among the places A,B,C,….., then the resources at the destination cannot exceed the threshold.

Vehicles :

The small vehicle is capable of carrying half the load compared to a large vehicle. The vehicles can carry a load equal to or less than their respective capacities. The vehicles can't store any excess resources, even if their capacity is more than the amount required for transportation.Vehicles can either travel horizontal or vertical only. Any number of vehicles are allowed in a path at a time.

When the condition in the provided pseudocode (given in update.pdf) is executed and the input data is updated , it's important to note that the locations of the vehicles remain unchanged. In other words, they remain at the positions where they travelled to in the previous iteration. So, the output should be updated according to the new data generated and the positions of vehicles at the end of previous iteration.This is to be done 3 times.

Give index to the vehicles such that small are labelled as S1,S2,... and so on while large vehicles are labelled as L1,L2,... and the order in which you label them is your choice.(based on these labels the output is to be written)

Input:

A file inputs.txt is provided.

The first line specifies the number of locations (k) where resource distribution occurs. It is in the following format:

No of locations: num

Subsequent lines provide information such as the threshold, the count of supply centres, the number of large and small vehicles at each supply centre and the capacity of a large vehicle.

This is followed by the information related to each location in each line, its format is as follows:

loc i: (x,y) resources: r large vehicles: l small vehicles: s

Subsequently, the particulars of each supply centre are outlined in individual lines as

supply i: (x,y) resources: r

Details of damage in each point are given by

damage(x,y): d

Each input is given in each line for ease of reading the file. The formats above are specified with exact spaces and format of how each line would be present for the same.

If any further specifications are required for you to read the input file, please mention them in your README file.

Output format:

Create a file “output.txt” with output as follows:

Iteration 1:

Vehicle 1: …

….

Vehicle i: (xi1,yi1) ->...->(xis1,yis1)(s,(xid1,yid1,ld1),(xid2,yid2,ld2),...)->...->(xid1,yid1)(d)...

…

Vehicle n: …

Print the entire trajectory for each vehicle, marking any point that acts as a source at a given instant as "source_coordinates(s, (destination1_coordinates, load_to_destination1), ...)" indicating the destinations it supplies. If a point is a destination, represent it as "destination_coordinates(d)".

Once all the vehicle paths are printed, provide a list of locations and the corresponding time required to replenish their deficiency. If a location is self-sufficient from the beginning then its corresponding time is considered 0.

If there is no valid path , print *"no path"* in the file .If the resources are not sufficient to fill up the deficiencies, print *"insufficient resources"* in the file. In case of printing time for these print "*couldn't reach*" in the file.

In place of vehicle i ,use the index fixed by you for the vehicle.

Example:

S1: (1,4)->(2,5)->...->(3,19)(s,(14,6,30),(3,6,50)) …->(14,6)(d)

…

L3: …

…

Destination: Time

Loc2: 30

…

This is repeated with each iteration. Their values are followed by the next line .(as shown below)

Iteration 1:

…..

Iteration 2:

…...

Iteration 3:

……

Input and output formats are better understood by referring to the samples provided.

Report:

Write a report explaining your approach to solve the problem. Mention assumptions clearly, if you have taken any.

Objectives :-

If T_i ; i = {1,2,....} are times taken to supply resources to locations A, B,….. respectively,

The main objective is to ensure that all the deficient places are supplied with sufficient resources in the least time possible. {i.e., make as minimum as possible}

NOTE :- The trucks travel a unit distance per unit time. The time taken for loading and unloading are ignored.

Distance travelled = Time taken (in terms of values)

Output :- The code should give a text file with route map.

Judging Criteria :- max of 'T_i', mean of 'T_i' & standard deviation of 'T_i'are considered to ensure that resources are supplied to all the deficient locations in the least time possible.
Note: If there is any destination which couldn't be filled up due to path or resource constraints it will not be considered in the above calculations of Ti.

How and what to submit

Submit README.txt,Report.pdf,output.txt and Source Code files in your github and submit your github link through the google form.