CS 218 DATA STRUCTURES

ASSIGNMENT 3- Fall 2023

***DUE**: Dec 1, 2023*

NOTE: Late submissions will not be accepted

***TO SUBMIT:** Documented and well-written structured code in C++ in the classroom. Undocumented code will be assigned a zero.*

Healthcare Relief Routing System

Problem Background

This assignment aims to design and implement a routing system that efficiently delivers healthcare commodities and equipment from the UN center to various hospitals in war zones, ensuring the optimal utilization of resources. Given the reachability information of all the hospitals and the UN center, the goal is to distribute healthcare supplies fairly among various hospitals according to the availability of the supplies and their demand by the respective hospitals.

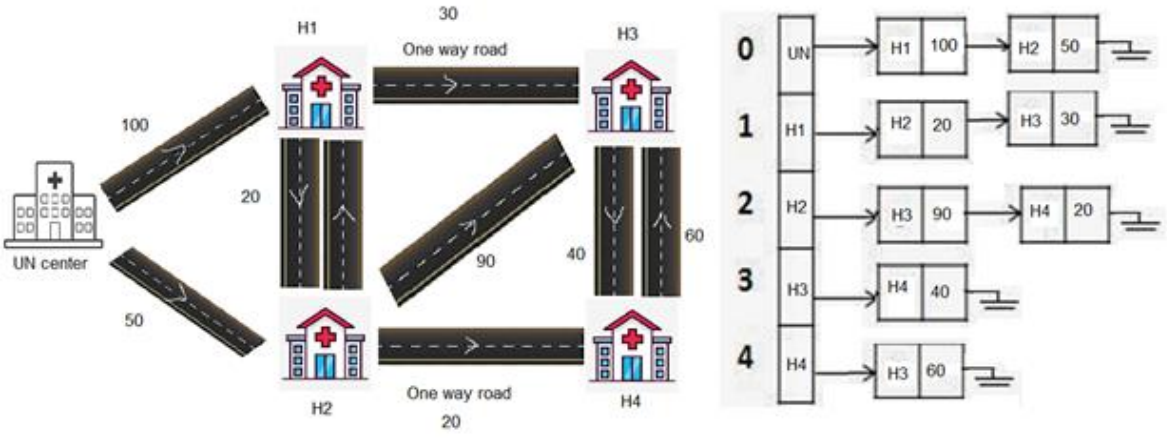


Figure 1 shows an example of a UN center and the Hospitals, along with the time (in minutes) needed to reach a hospital.

Requirements:

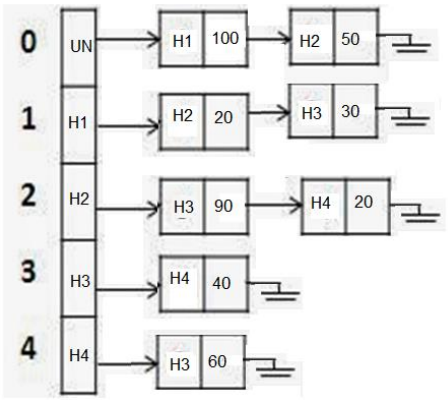
1. **Input Data:**
   - A file contains information on hospital locations and the time taken to reach the hospital from either the UN center or another hospital. You can assume that hospitals have IDs from 1 to the total number of hospitals, and the UN center has ID 0.
   - The format of the file is as follows:
     1. The first line states the total number of hospitals
     2. The second line gives information about the hospitals directly reachable from the UN center. The second line contains comma-separated pairs, where each pair is a hospital ID (directly reachable from the UN center) and the time needed to reach that hospital.
     3. The third line onwards contains hospital ID and comma-separated pairs where each pair includes the hospital ID and time needed to reach that hospital from the one mentioned at the start of the line.
   - Below is a sample input file (for the scenario shown in Figure 1)

4		
UN	**(H1,100)**	**(H2,50)**
1	**(H2,20)**	**(H3,30)**
2	**(H3,90)**	**(H4,20)**
3	**(H4,40)**	
4	**(H3,60)**	

Here, there are a total of 4 hospitals. Hospital 1 is reachable from the UN center in 100 minutes, whereas Hospital 2 is reachable from the UN center in 50 minutes. Also, Hospitals 2 and 3 can be reached from Hospital 1 in 20 and 30 minutes, respectively.

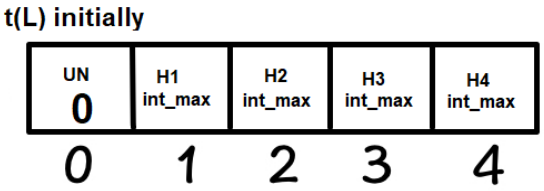
2. **Data Representation:**

Use a vector of a Linked list to store the reachability information from the file. The vector size must be 1+ the number of hospitals, and each element of the vector is a linked list to store ID and time pairs. ***Use the standard template library vector and List for this purpose.***



3. **Routing System Implementation:**
   - Implement a routing system that efficiently computes the optimal routes for delivering supplies from the UN center to each hospital. The optimal route is one that takes minimum time in minutes.

   - ***Use the following algorithm to compute the optimal paths.***
     1. Let **t[L]** represent the optimal time taken to reach **location L** from the UN center and **p[L]** represent the previous location in the optimal route to reach L. Initially, **t[L]** is intMAX if the location is a hospital and 0 if the location is UN center. Similarly, **p[L]** must be initialized to -1 (no previous location yet)



     2. Construct a priority queue (using min heap) on **t[L]** for all locations.
     3. Loop until the priority queue (heap) is empty.

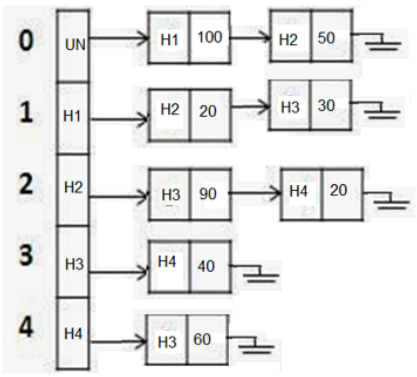
        i. Extract the location U such that t[U] is minimum from the heap
        ii. For each location V, that is reachable from U

        If t[V] is greater than (t[U] + time taken to reach V from U), Then, update t[V] = t[U] + time taken to reach V from U (this will require a decrease key operation in the heap)

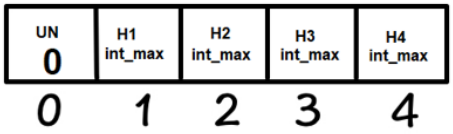
   - After the execution of these steps, t[L] must contain the minimum time to reach location L from the UN center. And **p[L]** can be used to compute the optimal path from the UN center to L

   - Use your implementation of a heap data structure to prioritize hospitals based on their travel times.

Example

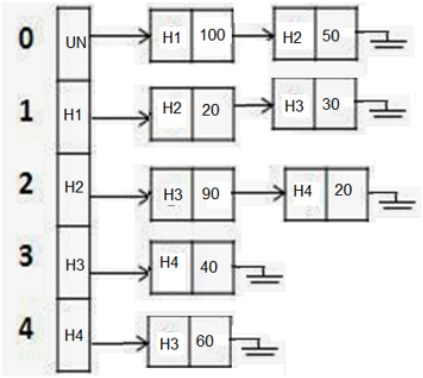


t(L) initially

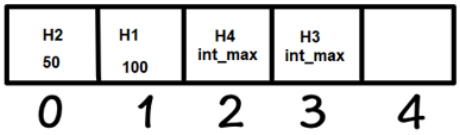


Construct Priority queue on t(L)
Extract min (that will be UN center)

As from UN we can reach H1 in 100m and H2 in 50m
So update the times in t(L) priority queue and
call decrease key function to keep it correct



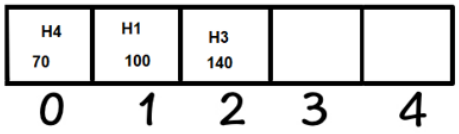
t(L)

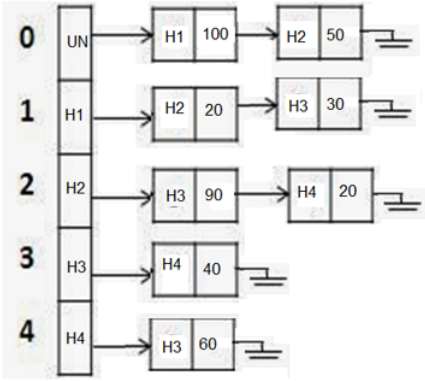


Extract min again (that will be H2)

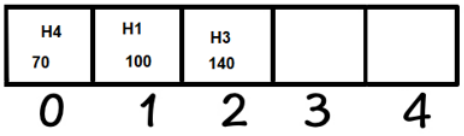
As from H2 we can reach H3 in 90m and H4 in 20m
So update the times in t(L) to 50+90 and 50+20 and
call decrease key function on heap

t(L)



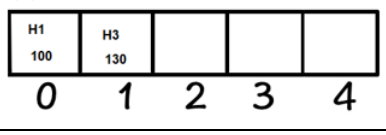


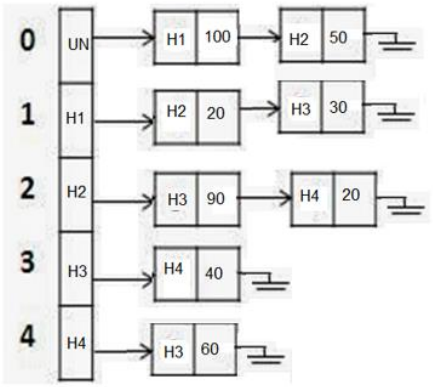
t(L)

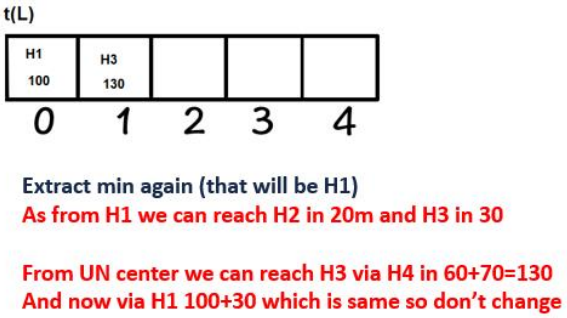


Extract min again (that will be H4)
As from H4 we can reach H3 in 60m
From UN center we can reach H3 via H4 in 60+70=130
As 130 is < 140 (previous time to reach H3)
So update the time in t(L) for H3 to 130 and
call decrease key function on heap

t(L)







Extract min again (that will be H1)
As from H1 we can reach H2 in 20m and H3 in 30

From UN center we can reach H3 via H4 in 60+70=130
And now via H1 100+30 which is same so don't change

And so on

You can also maintain **p[L]** to keep track of the previous location in the optimal route to reach L.

4. **User Interface:**
   - Develop a user interface to input the reachability file and display the optimal routes to reach each hospital.
   - Present the time taken for each route and the total time for delivering supplies to each hospital.

Submission:

- Submit the source code along with any necessary files.