

## Written Assignment 09

## WRITTEN ASSIGNMENT 09

Let  $G(V, E)$  be an *undirected* graph. Let  $w : E \rightarrow \mathbf{Z}$  be a function assigning integer weights to all the graphs edges and let  $r$  be the root vertex that will start to grow the minimum spanning tree (MST). Every vertex  $v \in V$  stores  $v.key$  – the key for a priority queue (initially containing all the vertices). A vertex also stores  $v.p$  – its parent (the parent vertex in the ultimate MST; it is assigned only once). Prims algorithm to find the minimum spanning tree in  $G$  is given by the following pseudocode:

```

MSTPRIM( $G, w, r$ ):
    for each vertex  $u \in V$ :
         $u.key = \infty$ 
         $u.p = \text{NULL}$ 
     $r.d = 0$ 
     $Q = \text{MINIMUMHEAP}(V)$     (Insert all vertices in a priority queue)
    while  $Q \neq \emptyset$ :
         $u = \text{EXTRACTMIN}(Q)$     (pick a vertex closest to the MST built so far)
        for each  $v \in \text{ADJ}(G, u)$ :
            if  $v \in Q$  and  $w(u, v) < v.key$ 
                 $v.p = u$ 
                 $v.key = w(u, v)$ 

```

Denote the last three digits of your Student ID by  $a, b, c$ . Student ID often looks like this: 201RDBabc, where  $a, b, c$  are digits. Compute three more digits  $x, y, z$ :

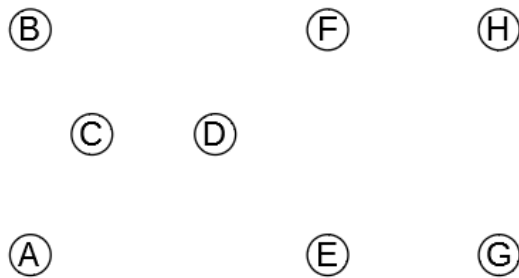
$$\begin{cases} x = (b + 4) \bmod 10 \\ y = (c + 4) \bmod 10 \\ z = (a + b + c) \bmod 10 \end{cases}$$

In this task the input graph  $G = (V, E)$  is given by its adjacency matrix:

$$M_G = \begin{pmatrix} 0 & 0 & 5 & 8 & y & 0 & 0 & 0 \\ 0 & 0 & 3 & 7 & 0 & z & 0 & 0 \\ 5 & 3 & 0 & 3 & 0 & 0 & 0 & 0 \\ 8 & 7 & 3 & 0 & 1 & 7 & 0 & 0 \\ y & 0 & 0 & 1 & 0 & 6 & 9 & 6 \\ 0 & z & 0 & 7 & 6 & 0 & x & 2 \\ 0 & 0 & 0 & 0 & 9 & x & 0 & 7 \\ 0 & 0 & 0 & 0 & 6 & 2 & 7 & 0 \end{pmatrix}.$$

- (A) Draw the graph as a diagram with nodes and edges. Replace  $x, y, z$  with values calculated from your Student ID. Label the vertices with letters  $A, B, C, D, E, F, G, H$  (they correspond to the consecutive rows and columns in the matrix).

If you wish, you can use the following layout (edges are not shown, but the vertex positions allow to draw the edges without much intersection). But you can use any other layout as well.



- (B) Run Prim's algorithm to find MST using  $r = A$  as the root. If you do not have time to redraw the graph many times, just show the table with  $v.key$  values after each phase. (No need to show  $v.p$ , as the parents do not change and they are easy to find once you have the final rooted tree drawn.) The top of the table would look like this (it shows Phase 0 – the initial state before any edges have been added).

Phase	A	B	C	D	E	F	G	H
0 (initial state)	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$

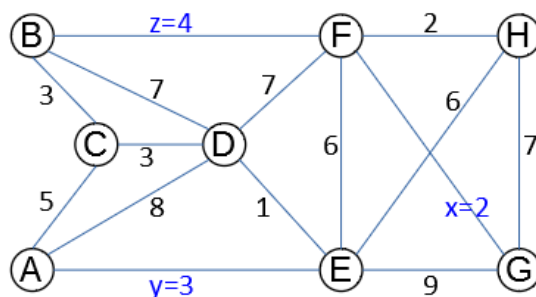
- (C) Summarize the result: Draw the MST obtained as the result of Prim's algorithm, find its total weight.

**Answer:**

- (A) As an example, consider Student ID with these last 3 digits:  $(a, b, c) = (7, 8, 9)$ . Compute the values of  $x, y, z$ :

$$\begin{cases} x = (b + 4) \bmod 10 = 2, \\ y = (c + 4) \bmod 10 = 3, \\ z = (a + b + c) \bmod 10 = 4. \end{cases}$$

Let us draw the graph with edges (including those labeled by  $x = 2, y = 3, z = 4$ ).



- (B) We show the values of vertices in the priority queue (the value shows the minimum distance to some vertex in the tree built so far using Prim's algorithm). The vertices that have been extracted (removed) from the queue show – instead of the key value.

Phase	A	B	C	D	E	F	G	H
0 (initial state)	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1 (extract A)	—	$\infty$	5	8	3	$\infty$	$\infty$	$\infty$
2 (extract E)	—	$\infty$	5	1	—	6	9	6
3 (extract D)	—	7	3	—	—	6	9	6
4 (extract C)	—	3	—	—	—	6	9	6
5 (extract B)	—	—	—	—	—	4	9	6
6 (extract F)	—	—	—	—	—	—	2	2
7 (extract G)	—	—	—	—	—	—	—	2
8 (extract H)	—	—	—	—	—	—	—	—

**Note:** During Phase 7 there are two vertices of the same priority value ( $G$  and  $H$ ). We select  $G$  which is alphabetically earlier and add the edge  $(F, G)$  to the MST. In this situation the order how edges  $(F, G)$  and  $(F, H)$  does not matter much. In some other graphs the choice of edges (for equal keys in the priority queue) can lead to major differences in subsequent steps – and there may exist multiple MSTs.

- (C) Every time we extract a vertex from the priority queue in (B), we set its parent to the other end of the edge that was safe to add at this step. Below is the picture of the resulting MST (tree edges shown bold and light-gray, all the other edges are shown as dashed).

