United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Final Exam   Total Marks: **40**   Summer 2023

Course Code: CSE 2217   Course Title: Data Structure and Algorithms II

Time: 2 hours

Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.

There are **six** questions. **Answer all of them**. Show full simulation/tabulations wherever necessary. Figures in the right-hand margin indicate full marks.

1. **(a)** Find the Minimum Spanning Tree (MST) for the given graph in Figure 1 using **Prim's algorithm**. Show the details of your calculation. **Explain why** your MST satisfies the properties of minimum spanning tree. **[4]**

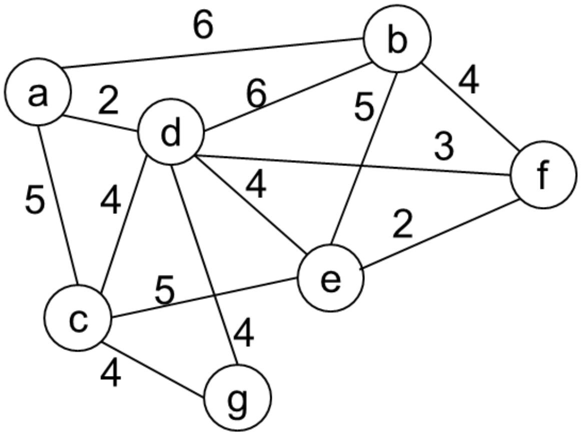


Figure 1: An undirected graph G(V,E) for **Question 1(a)**

(b) Show, through an example, that the following statement is **true**: "*For a directed graph, Kruskal's algorithm fails to generate MST.*" **[2]**

2. **(a)** What is a rolling hash and how does it help in efficiently updating the hash value in the context of the Rabin-Karp algorithm? **[1]**

(b) Why is it necessary to use the modulo operation in the hash function of the Rabin-Karp algorithm? **[1]**

(c) You are given a DNA sequence and a pattern. The DNA sequence is represented by a series of nucleotides (A, C, G, and T) which means the sequence contains only the letters A, C, G, and T. You need to figure out the number of occurrences of the given pattern in the DNA sequence using **Rabin-Karp algorithm**. You must show the Hash values and calculations for all the substrings. **]3]**

The numeric values of each nucleotide are: A = 1, C = 2, G = 3, and T = 4.

Sequence = **AGCTAGCGAGCTAG**
Pattern = **AGCTAG**

The hash function is as follows:
hash(s) = [{s[0] * 4^(n-1)} mod 7 + {s[1] * 4^(n-2)} mod 7 + ……. + {s[n-1] * 4^(n-n)} mod 7] mod 7
where,
hash(s) = hash value of string s
n = length of the string s

3. **(a)** Can you calculate the shortest path from vertex A to vertex E in the following graph? If yes, mention the name of the algorithm you will use. If no, write down why. **[2]**

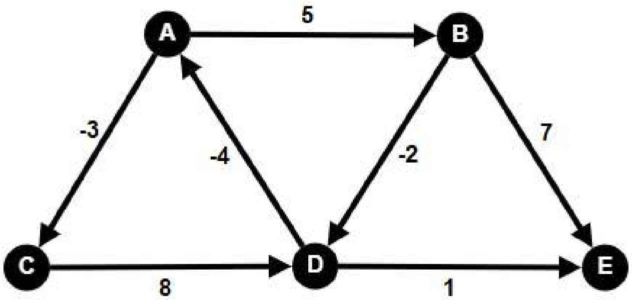


Figure 2: A directed graph G(V,E) for **Question 3(a)**

(b) Find out the shortest path from vertex A to vertex G in the following graph using **Dijkstra's** shortest path algorithm. **Show the calculations in detail** and **write down** the vertices in the shortest path along with the **path length**. **[5]**

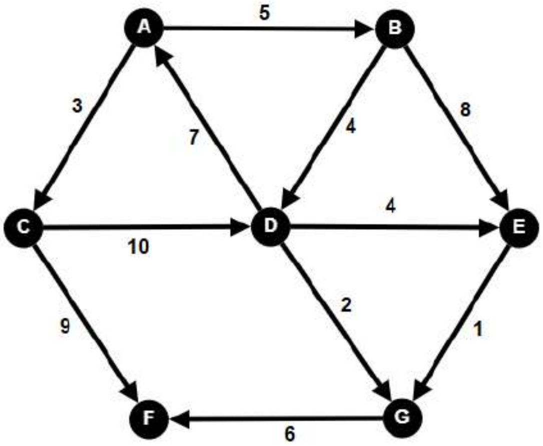


Figure 3: A directed graph G(V,E) for **Question 3(b)**

(c) Find out a topological ordering of the following directed acyclic graph. You may use Breadth First Search (BFS) or Depth First Search (DFS) to find the order. **[3]**

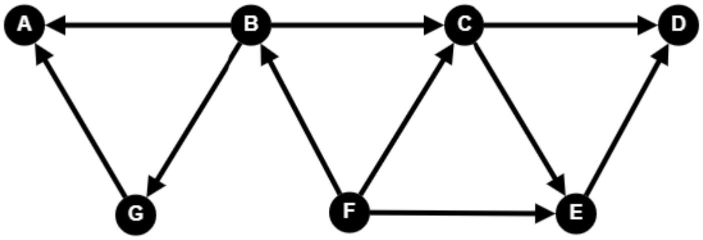


Figure 4: A directed acyclic graph G(V,E) for **Question 3(c)**

4. **(a)** The following table shows the parent array of a Disjoint set (Rooted tree implementation). Perform the following operations sequentially using **path compression and union-by-rank heuristic:**

i. Draw the disjoint set forest **[2]**
ii. Redraw the forest after Union(3,7) **[1]**
iii. Redraw the forest after Union(5,10) **[1]**
iv. What will be returned by Find-Set(9)? **[1]**

Index	0	1	2	3	4	5	6	7	8	9	10	11
Parent	0	0	1	2	2	4	4	0	8	8	9	8

(b) During the execution of CONNECTED-COMPONENTS on an undirected graph G = (V, E) with k connected components, how many times is **FIND-SET** called? How many times is **UNION** called? **Express your answers in terms of |V|, |E|, and k.** **[2]**

```
CONNECTED_COMPONENTS(G)
  for each vertex v in V[G] do
   MAKE_SET (v)
  for each edge (u, v) in E[G] do
   if FIND_SET(u) != FIND_SET(v) then
      UNION(u, v)
```

5. **(a)** Consider an open-addressing hash table as shown below. The table already contains four data items, and other empty slots contain NIL. Assume that collisions are handled using the hash function

$$\boldsymbol{h(k, i) = (h'(k) + i\, h_2(k))} \text{ mod } \mathbf{13}.$$

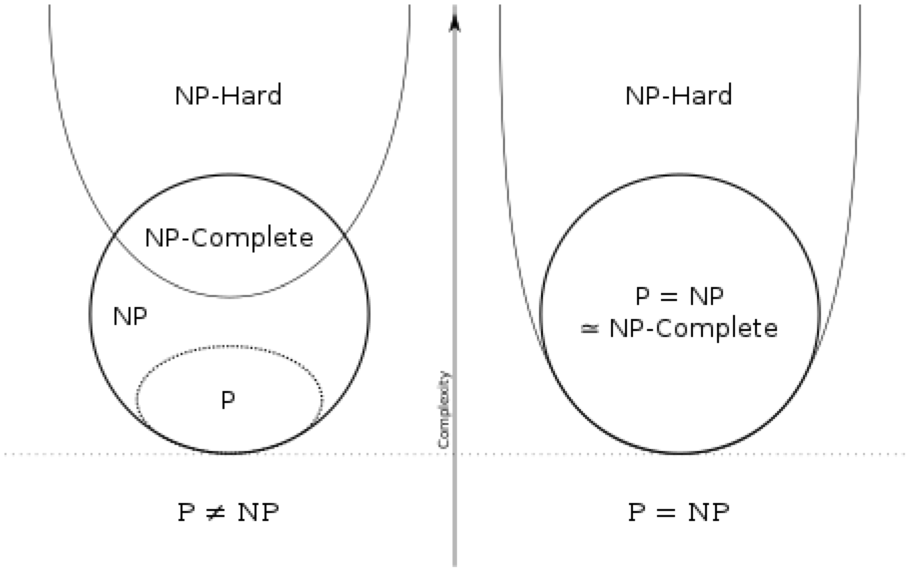
$$\text{where } \boldsymbol{h'(k)} = (3\boldsymbol{k} + 5) \text{ mod } \mathbf{13},$$

$$\text{and } \boldsymbol{h_2(k)} = (5\boldsymbol{k} -12) \text{ mod } \mathbf{13}.$$

By showing detailed calculations, redraw the table after

(i) *insert* 89; **[2]**
(ii) *insert* 55. **[2]**

0	1	2	3	4	5	6	7	8	9	10	11	12
	20					39	60			45		

	(b) What is primary clustering? What can we do to avoid primary clustering in hash table? **(c)** Suppose given a hash table of size 10, you are asked to insert 13 keys in the table. How can you solve this? Explain.	**[2]** **[1]**
6.	**(a)** What is the difference between Deterministic and Non-deterministic Algorithms? Explain briefly with proper examples. **(b)** When does a problem belong to the complexity class NP? How does it differ from P-class problems? **(c)** What does the following diagram represent? Explain briefly.  NP-Hard NP-Complete NP P Complexity NP-Hard P = NP ≃ NP-Complete P ≠ NP P = NP Figure 5: for **Question 6 (c)**	**[1]** **[2]** **[2]**