United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Final Exam Total Marks: **25** Summer-2021

Course Code: CSI 227/CSE2217 Course Title: Data Structure and Algorithms II

Time: *1 hour 15 minutes* for answering. *Another 15 minutes* for download and upload

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

There are FOUR questions. Answer all of them. Figures in the right-hand margin indicate full marks.

You must use the values of x and y in Figure 1 and Question 1, 2, 3 and 4.

First determine x and y correctly for your student ID

For example, a student with ID: **011 142001**

A B

A=20, **B**=1

x = 1 + (A mod 6) = 1+(20 mod 6) = 3

y = 1 + (B mod 6) = 1+(1 mod 6) = 2

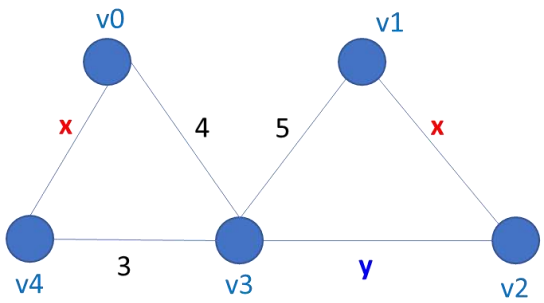

Figure 1: An undirected graph G(V,E)

1 **(a)** Find the MST for the given graph in Figure 1 using Kruskal's algorithm. Show the details of your calculation. **[3]**

(b) What will be the runtime complexity of Kruskal's algorithm if we use bubble sort to sort edges (Sorting n elements with bubble sort takes time $O(n^2)$) ? **[1]**

2 **(a)** Find the *Shortest path tree* for the given graph in Figure 1 using Dijkstra's SSSP algorithm, where the source is vertex **v0**. Show the details of your calculation. **[5]**

(b) A graph contains the vertices {A, B, C, D, E, F, G} and the shortest path from A to B is A→E→C→D→F→G→B. Is it possible to find the shortest path from E to F from given data? Justify your answer. **[2]**

3 **(a)** Insert the following six values sequentially for a hashtable with open-addressing that uses linear probing: 17, 23, 21, **y**, 75, 29. **[5]**
Consider the hash function to be: **h(k) = (k + x) mod 7** and hashtable size to be 7.
Also, report the length of the longest primary cluster after completing all insertions.
You must show the hash values for each key and the final hash table.

(b) Perform the following operations for a disjoint set *sequentially*. Assume the elements of the disjoint set are 0 to 7. Apply *union-by-rank* and *path-compression* where applicable. **[4]**

(i) Makeset(0)
(ii) Makeset(1)
(iii) Makeset(2)
(iv) Makeset(3)
(v) Makeset(4)
(vi) Makeset(5)
(vii) Makeset(6)
(viii) Makeset(7)
(ix) Union(**x**, **y**)
(x) Union(**y**, **y** +1)
(xi) Draw the disjoint set forest.
(xii) Find_set(**x**)

4 **(a)** Consider the following code for solving a decision problem named "*Kryptonite*": The *read_input* function runs in **O(n)** time for the input size n whereas the *reduce* function runs in $\boldsymbol{O}(\boldsymbol{n}^{\boldsymbol{x}})$ and the *decision_solver* function runs in $\boldsymbol{O}(\boldsymbol{n}^{\boldsymbol{y}})$ time. Does the problem belong to class P? Explain your answer briefly. **[2]**

```
def main():
    x = read_input()
    x_transformed = reduce(x)
    x_solved = decision_solver(x_transformed)
    print(x_solved)
```

(b) Briefly explain the differences between the complexity classes NP-Complete & NP Hard. **[2]**

(c) Consider a NP Hard problem, P has been reduced to another problem Q in $O(1.5^n)$ time. Now in future, an algorithm has been found that solves the problem Q in polynomial time. Will that imply $P = NP$ then? Explain the answer briefly. **[1]**