United International University
CSI 227: Algorithms, Fall 2017
Final Exam
Total Marks: 120, Time: 2 hours

Answer all the questions from section A, and any 3 out of the 5 in section B. ($6 \times 20 = 120$).

Section A

1. (a) Why is the space-complexity of the **adjacency-list** representation of an undirected graph is $\mathcal{O}(n + m)$? What would the complexity be if the graph is directed instead? [2 + 2]

   (b) Use the **Disjoint-Set Forests** data structure to find the number of connected components in the graph $G = (V, E)$ where $V = \{a, b, c, \cdots, i\}$ and $E = \{(a, b), (b, c), (c, a), (d, e), (e, f), (e, g), (f, d), (g, h), (e, h)\}$, using the **union-by-rank** and the **path-compression** heuristics. [9]

   (c) Consider the **Kruskal**'s algorithm provided at fig. 1. For a connected undirected graph G, how many times in total will the $FIND - SET$ and the $UNION$ procedures get called? If you use **insertion sort** to sort the edges, each $FIND - SET$ takes $\mathcal{O}(n)$, and each $UNION$ takes $\mathcal{O}(1)$ time, what will the total running-time be? [4 + 3]

```
KRUSKAL(G):

    let G = (V, E)

    sort E in non-decreasing order of weigh
    let T be an empty set

    for each edge e from E in sorted order
        let e = (u, v)

        set_u = FIND-SET(u)
        set_v = FIND_SET(v)

        if set_u != set_v
            add e to T
            UNION(set_u, set_v)

    return T
```

Figure 1: Q. 1c

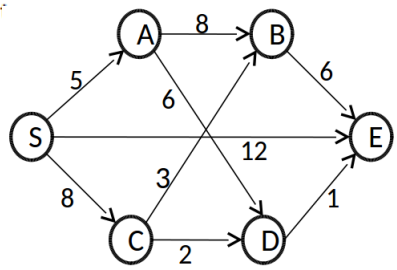

Figure 2: Q. 2b

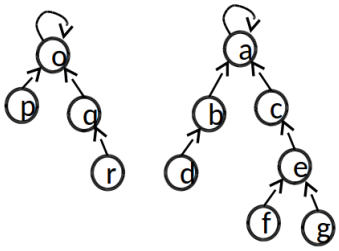

Figure 3: Q. 4a

2. (a) Provide an $\mathcal{O}(n)$ algorithm that detects whether a provided vertex v is isolated (not connected to any other vertex) or not. Assume that the graph $G = (V, E)$ is represented in the **adjacency matrix** form. [6]

   (b) i. Fig. 2 shows the bus routes of 6 cities. Find the shortest distances from city A to all other cities. [8]

   ii. If we increase the weight of all edges by 5, will the routes change? If so, which one will change? [6]

3. (a) What is the **simple uniform hashing** assumption? Consider the **chained hashing** scheme. With simple uniform hashing, what will be the expected cost for searching a data item? Without this assumption, what will be the cost for insertion of data items? Explain very briefly. [4 + 3 + 3]

   (b) At the **Rabin-Karp** algorithm for string matching, what will be the problem if the hash values of the substrings are **not reused** each time? [6]

   (c) Does the **sorting problem** belong to the class ***NP***? Why or why not? [4]

Section B

4. (a) What would the resultant forest be after calling $UNION(p, e)$ on the disjoint-sets forest in Fig. 3? You must use the **union-by-rank** and the **path-compression** heuristics for such. [8]

(b) Suppose that for a graph G, you already have found an MST T. Now a new edge e has been added to the graph. Describe how to find an MST of the new graph, using the already found MST T. [6]

(c) Give the best-case and the worst-case examples for the **naive string matching** algorithm where text size $n = 10$ and pattern size $m = 4$. [6]

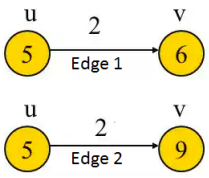

Figure 4: Q. 5a

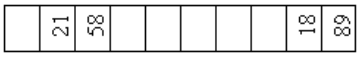

Figure 5: Q. 8a

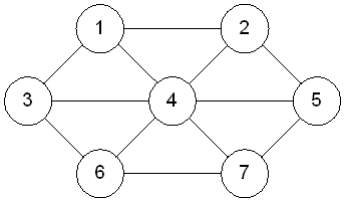

Figure 6: Q. 8b

5. (a) Consider the edges at Fig. 4. The shortest-path estimation of each vertex is shown within the vertex. What will be the new estimations after these edges are **relaxed**? [6]

(b) Problem X has an algorithm with worst-case running time of $O(2^n)$. Is $X \in NP$? Why or why not? [6]

(c) Suppose that using some unknown algorithm, you have obtained the following shortest path from a source s to a destination t : s, d, f, g, f, r, t. The graph contains all positive edge-weights. Even though you do not know what algorithm has been used, mention why is this shortest path incorrect. [8]

6. (a) Assume that the alphabet set $\Sigma = \{a, b, c, d, e, f\}$ has $d = 6$ characters. Demonstrate what happens when we insert the keys $\{ff, ee, db, cd, ca, cf, be\}$ into a **chained hash table**. Let the table have 9 slots, the hash function be $h(k) = k \bmod 9$, and the characters in Σ have values as $0, 1, \ldots, 5$ respectively. [9]

(b) Briefly explain with examples when and how the **union by rank** heuristic prevent the increase of ranks of disjoint-set trees at the $UNION$ operations. [6]

(c) Provide an example of a problem that belongs to the class $\boldsymbol{P}$, and explain very briefly why it is in $\boldsymbol{P}$. [5]

7. (a) For the **SSSP** problem, consider a directed graph with n vertices and m edges. Answer the following.

i. How many times each edge is relaxed in the **Dijkstra**'s algorithm? How many times for the **Ballman-Ford** algorithm? [4]

ii. Suppose that you are using just a simple array for the distances at the **Dijkstra**'s algorithm. What will be time cost to pick each vertex? What is the cost of an edge relaxation? [4]

iii. If each vertex could be picked in $\mathcal{O}(1)$ time, and each edge relaxation took $\mathcal{O}(1)$ time, what would have been the running-time of the **Dijkstra**'s algorithm? How? [4]

(b) Consider the following text $T =$ "$bcedja$" and pattern $P =$ "$cedj$". Suppose that the alphabet consists of just the $d = 10$ letters $\{a, b, c, ..., i, j\}$. Using modulo $q = 13$, find out how many **valid matches** and **spurious hits** will the **Rabin-Karp** algorithm find. Use $a = 0, b = 1, ...$ for hashing. [8]

8. (a) Consider the open-addressing hash table T in Fig. 5 with size $m = 10$. The table contains 4 data items already, and other empty slots contain *null*.

i. You are using the hash function $h(key, i) = (key + i^2)\ mod\ m$ for. Show the table state after operations i - iii, and output for iv: (i) insert 68; (ii) insert 7; (iii) delete 58 (replacement with *null*); (iv) search 7. [8]

ii. Although 7 had been present at the hash table, your search would fail here. Mention another such search call that will fail even if the key is present. How may you modify the operations such that this will be prevented? [2.5 + 2.5]

(b) Consider the graph at figure 6. Assign weights to each of the edges such that the **MST** of this graph consists exactly of the edges $(4, 1), (5, 4), (3, 4), (4, 6), (4, 7)$, and $(2, 4)$. What is the **MST** cost? [6 + 1]