United International University
Department of Computer Science and Engineering
CSI 227: Algorithms, Final Exam, Spring 2017
Total Marks: 120, Time: 2 hours

Answer any 6 out of 8 questions ($6 \times 20 = 120$).

1. (a) For a graph $G = (V, E)$ represented in a simple adjacency list representation: [3+4+5]

   i. Derive the space-complexity of representing the graph

   ii. Derive the runtime complexity for finding out the maximum degree of G

   iii. Provide an algorithm to find the **in-degree** of all the vertices if G is directed. Derive the running-time of your algorithm. The in-degree of a vertex is the number of edges terminating at that vertex.

   Explain your derivations very briefly.

   (b) Show the values for *distance* (d) and *parent* (π) that result from running Breadth-First-Search (BFS) on the following undirected graph, using vertex c as the source vertex. [4]

   (c) "Every connected directed acyclic graph (DAG) has exactly one topological ordering" - true or false? Design a graph G with exactly 4 vertices that justifies your reasoning. [4]

2. (a) Would the following edges always be a part of a Minimum Spanning Tree (MST) of a graph G containing 10 vertices when the edge weights are guaranteed to be unique? Explain your answer.
   Edge with the (i) smallest weight, (ii) second smallest weight, and (iii) third smallest weight. [8]

   (b) If we do not follow the *union-by-rank* heuristic at the $Union(x, y)$ algorithm for a disjoint-sets data structure, what can be the maximum height of a final disjoint-set tree with n elements, resulting after $n - 1$ operations? Describe with an example with 8 elements and necessary $Union$ operations. Then incorporate the *union-by-rank* heuristic for the exact same elements and $Union$ operations, and find out the tree height. [2+3+4]

   (c) Is it possible to implement Kruskal's algorithm without disjoint-sets data structure? How so? [3]

3. (a) Suppose that you have performed the Dijkstra's algorithm to find the shortest paths from a single source in a given graph G with positive edge weights. If all the edge weights are reduced by the same constant amount, would you need to perform the Dijkstra's algorithm again in order to find the shortest paths? Why or why not? Illustrate your answer with example(s). [10]

(b) Given in Fig. 1(a) is Dijkstra's algorithm to the *single source shortest path* problem. What would be the time complexity of the algorithm for an input graph G=(V,E) if (i) an array is used instead of priority queue (ii) adjacency matrix is used instead of adjacency list. Briefly justify your answers. [5+5]

```
DIJKSTRA(G,w,s):
    let G=(V,E)
    INITIALIZE-SINGLE-SOURCE(G,s)
    Q = G.V
    while Q is not empty:
        u = EXTRACT-MIN(Q)
        for each v in adjacency_list[u]:
            RELAX(u,v,w)

RELAX(u,v,w):
    if distance[v] > distance[u] + w(u,v):
        distance[v] = distance[u] + w(u,v)
        parent[v] = u
```

(a) Q. 3(b)

```
BFS(G,s):
    let G=(V,E)
    for each v in V:
        visited[v] = 0, parent[v] = null
        distance[v] = INF
    visited[s] = 1, distance[s] = 0
    declare an empty queue Q
    ENQUEUE(Q,s)
    while Q is not empty:
        u = DEQUEUE(Q)
        for each v in adjacency_list[u]:
            if visited[v] == 0:
                visited[v] = 1, parent[v] = u
                distance[v] = distance[v] + 1
                ENQUEUE(Q,v)
```

(b) Q. 8(a)

Figure 1: •

4. (a) Suppose you have a hash table T with $m = 6$ slots that uses open addressing with the hash function h(k,i) = (k mod m + i) mod m. Now perform the following operations on T assuming all slots have NIL when you begin. Show the state of T after each operation clearly marking the value at each slot.
(i) INSERT(T,27) (ii) INSERT(T,15) (iii) INSERT(T,39) (iv) DELETE(T,15) (v) SEARCH(T,39) [2x5]

   (b) What would be the worst case time complexity of Rabin-Karp algorithm if we have to recompute the hash values everytime for every m-length substring? [5]

   (c) You are given a problem A that is solvable in polynomial time. How can you use this property of problem A to prove that another problem **B** is also solvable in polynomial time? [2]

   (d) A problem X has an algorithm with the worst case running-time $O(n!)$. Is $X \in NP$? Explain briefly. [3]

5. (a) For Direct-Address Tables, what are the benefits of storing data items outside of the table, and storing pointers / references to those data items at the table? [3]

   (b) Given that memory is not an issue to be concerned, what is the best possible hashing scheme? Why? [2]

   (c) The diameter of graph is the"longest shortest path" i.e. to find the diameter of a graph, first you have to find the shortest path between each pair of vertices. The greatest length of any of these paths is the diameter of the graph. A disconnected graph has infinite diameter. Now provide $O(n(m+n))$ algorithm to find the diameter of an unweighted graph G=(V,E) where $|V| = n$ and $|E| = m$. [10]

   (d) Kruskal's algorithm may provide different Minimum Spanning Trees (MST) based on how it sorts the edge-list of the graph. Provide an example of a graph where the Kruskal's algorithm will return two different MSTs on two different runs of the algorithm, based on the edge-list being sorted differently. Explain your example briefly. On what condition(s) is a unique MST guranteed? [4+1]

6. (a) When does a problem belong to the complexity class NP? Why does the problem of finding a Hamiltonian cycle in an undirected graph belong to the complexity class NP? Explain in detail. [1+4]

   (b) Under what assumptions can we expect $\theta(1+\alpha)$ bound for search operations on chained hashing? [2.5]

   (c) The self-proclaimed *theoretical CS guru* Mr. Shan has got a map of a random country, and wants to visit the location t from the location s. Using his own shortest path algorithm, he designed the following

shortest path from s to t (each road costs him a positive amount of money):

$$s, q, e, f, h, u, a, k, p, z, c, b, m, e, n, b, t$$

Without even asking for the route costs or any other information, Mr. Imtiaz rejected the path, claiming it's not a shortest path from s to t. What logical reasoning did he use for the decision? [2.5]

(d) Working modulo $q = 11$ and $|\Sigma| = 10$, how many spurious hits does the Rabin-Karp matcher encounter in the text $T = 3141592653589793$ when looking for the pattern $P = 26$? [10]

7. (a) Provide a pseudocode for PRINT-SET(x), which is given a node x prints all the members of x's set. Assume that you have other Disjoint-Set operations (MAKE-SET, FIND-SET, UNION) at your disposal. What would be the complexity of your algorithm? [6+1]

(b) What will be the relation between complexity classes P and NP if a polynomial time algorithm can be devised for some NP-Complete problem? [3]

(c) There has been a magical explosion at one of the isles of the Skellige Islands, and legendary witcher Geralt is going to investigate it. Skellige consists of n islands, connected by **one-way** boat routes. For using these routes, one has to pay various amounts of Florens. Geralt is out of Skellige right now. With some help, he can teleport directly into any of the islands belonging to the set - $\{i_0, i_1, i_2, ..., i_k\}$. Representing the Skellige Islands with a graph, how can you efficiently find out the island to which he must teleport to, so that he can visit the explosion site, paying the minimum possible cost along his way? [7.5]

(d) State one advantage and one disadvantage of the Open Addressing scheme for hash tables over the chained hashing method. [2.5]

8. (a) A simple version of the *Breadth-First Search (BFS)* algorithm is given in Fig. 1(b), run from a vertex s on a simple undirected graph $G = (V, E)$. Modify the algorithm in a manner such that it will detect whether G contains cycle(s) or not. Assume the graph to be connected. [5]

(b) Suppose you want to modify the chaining scheme for Hash Table T to keep each list sorted. How does your modification affect the runtime of INSERT(T,x), SEARCH(T,k) and DELETE(T,x) operations? [5]

(c) Design a directed, weighted graph $G = (V, E)$ with 6 vertices ($V = \{s, t, u, v, w, x\}$) and 12 edges that contains negative-weight cycle(s), all the vertices in the graph has well-defined shortest paths from source s. A well-defined shortest path is a path with path cost $\neq -\infty$. [3]

(d) Use the disjoint-sets data structure with union-by-rank and path-compression to identify the number of connected components in the graph $G = (V, E)$ where $V = \{v_1, v_2, \ldots v_8\}$ and $E = \{(v_1, v_2), (v_2, v_3), (v_5, v_6), (v_4, v_7), (v_6, v_7), (v_6, v_8)\}$. Inspect edges in the order they appear in E in your simulation: [7]