Essential Algorithms **Name:** _____
CISC 233 Spring 2025
Exam 02
04/17/2025
Time Limit: 90 Minutes Mina Gabriel

Short Time Questions 25 - points

1. (5 points) What type of input gives the best-case performance for insertion sort? What is the best-case time complexity?

2. (5 points) Let H be a universal family of hash functions from a universe U onto a table of size m. Then $\Pr_{h\sim H}[h(e) = i] = 1/m$ for all elements $e \in U$ and table positions $i \in \{1, 2, \ldots, m\}$. Is this true or false? Why?

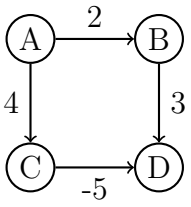
3. (5 points) Draw a recursive tree of the worst case for QuickSort. What is the time complexity?

4. (5 points)

$$\sum_{i=1}^{n}\sum_{j=1}^{k} 1 = O(n^2)$$

   True or False? If True, show why. If False, what is the correct asymptotic bound?

5. (5 points) Is it valid to use **Dijkstra's algorithm** on the following graph to find the shortest paths from node A to all others? If not, explain **why not**.



Medium Time Questions 30 - points

1. (10 points) Suppose you have been given an algorithm that can find the median in $O(n)$, if you use this algorithm for QuickSort,
   (a) (5 points) What is the time complexity in this case?
   (b) (5 points) If a pivot splits the list evenly, the recursive relationship is defined as $T(n) = 2T(\frac{n}{2}) + O(n)$. What is the recursive relationship in this case?

   Discuss your answer.

2. (10 points) When would you use BFS over DFS, and when would DFS be preferred instead? Explain with examples.

3. (10 points) If all the elements in an array are sorted in decreasing order and we want them in increasing order, which sorting algorithm will be asymptotically the fastest: *insertion sort, quick sort, merge sort*? Justify your answer with the appropriate time complexity and describe the impact of the input structure on each algorithm's behavior

Long Time Questions 45 - points

1. (15 points) You are implementing a hash table where each bucket uses a **Binary Search Tree (BST)** instead of a linked list. The hash function is defined using the **multiply-shift method**:

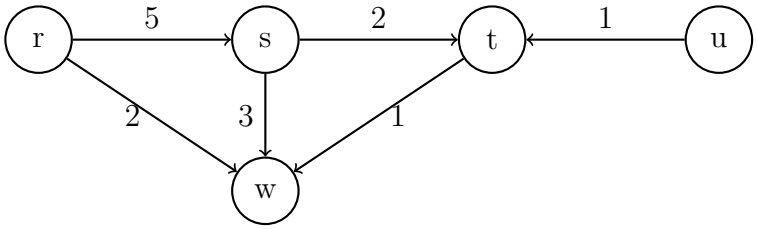
$$h_a(k) = ((k \times a) \bmod 2^w) \gg (w - \ell)$$

Assume the hash table is represented by an array T, and that $T[h_a(k)]$ gives the root of a BST at that index.

   (a) (10 points) Write pseudocode to search for a key k in the hash table. Define all variables you use. Your pseudocode must clearly show:
   - How the hash index is computed
   - How the BST is searched

   (b) (5 points) What is the time complexity of searching in this structure?

2. (15 points) The following directed graph represents a weighted network of nodes. Apply **Dijkstra's Algorithm** to compute the shortest-path distances from the source node r to all other nodes.



   (a) (10 points) Run Dijkstra's algorithm starting from node r. For each node $v \in \{r, s, t, u, w\}$, provide:

   - $d[v]$: the shortest-path distance from r to v
   - $\pi[v]$: the predecessor of v in the shortest-path tree (use `NIL` if none)

   Use ∞ for unreachable nodes.

   (b) (2 points) Draw the **shortest-path tree** rooted at r, showing only the edges used in the final shortest paths.

   (c) (2 points) What is the time complexity of Dijkstra's algorithm using a binary min-heap priority queue? Express it in Big-O notation in terms of $|V|$ and $|E|$.

   (d) (1 point) If the graph contains $|V| = 1000$ nodes and $|E| = 5000$ edges, estimate the worst-case number of operations (asymptotically) required to run Dijkstra's algorithm.

3. (15 points) **Dutch National Flag Problem (Three-color sorting)**

   You are given an array $A[1 \dots n]$ containing elements colored either **red (0)**, **white (1)**, or **blue (2)**, arranged in arbitrary order. Your task is to sort the array in linear time $O(n)$ such that all **red (0)** elements come first, followed by all **white (1)** elements, and finally all **blue (2)** elements. Explain your ideas in English or write a pseudocode to solve the problem.

   Example:

   $$\text{Input: } A = \{0, 1, 1, 0, 1, 2, 1, 2, 0, 0, 0, 1\}$$

   $$\text{Output: } A = \{0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2\}$$

   Explanation: The sorted array clearly groups all **red (0)** elements first, then all **white (1)** elements, followed by all **blue (2)** elements.