Algorithms
Comprehensive Exam
(Fall 2021)

SHORT QUESTIONS **(Answer all six questions, each carrying 7 points.)**

1. What is the respective time complexity expressions of the following two code segments?

a)

```
int count = 0;
for (int i = 0; i < n; i++)
   for (int j = 0; j < i; j++)
      count++;
```

b)

```
int count = 0;
for (int i = n; i > 0; i /= 2)
   for (int j = 0; j < i; j++)
      count++;
```

2. Use a hash table with size $m = 7$ to store the set of keys: $\{1, 5, 3, 9, 8, 16, 10\}$.

   a) Pick hash function $h(k) = k \ mod \ 7$ to hash. Please draw the table with keys hashed to each entry. If there exists collision in one entry, please chain the collided keys.

   b) Pick hash function $h(k) = k \ mod \ 7$ to hash. Please draw the table with keys hashed to each entry. If there exists collision in one entry, please use linear hashing method with probe sequence $h(k, i) = (h(k) + i) \ mod \ 7$ to handle collision.

   c) Pick hash function $h(k) = k \ mod \ 7$ to hash. Please draw the table with keys hashed to each entry. If there exists collision in one entry, please use double hashing method with probe sequence $h(k, i) = (h_1(k) + i * h_2(k)) \ mod \ 7$ to handle collision, where $h_1(k) = h(k) = k \ mod \ 7$, and $h_2(k) = k \ mod \ 5 + 1$.

   d) Which method gives the best performance? And which method gives the worst performance? Please give your explanation.

   e) For the double hashing method above, can we change $h_2(k) = k \ mod \ 5 + 1$ to $h_2(k) = k \ mod \ 5$? Please give your explanation.

3. Circle (T)rue or (F)alse in each of the following statement. No justification is needed.

   a) T F Given a connected graph G = (V, E), if a vertex v ∈ V is visited during level k of a breadth-first search from source vertex s ∈ V, then every path from s to v has length at least k.

   b) T F Every problem in NP can be solved within exponential time.

c) T F Given a weighted directed graph G = (V, E, w) and a shortest path p from s to t, if we doubled the weight of every edge to produce G* = (V, E, w*), then p is still a shortest path from s to t in G*.

d) T F Under the simple uniform hashing assumption, the probability that three specific data elements (say 1, 2 and 3) hash to the same slot (i.e., h(1) = h(2) = h(3)) is $\frac{1}{m^2}$, where m is the number of slots.

e) T F The following array is a max heap: [8, 4, 6, 1, 5, 2].

f) T F Every directed acyclic graph has exactly one topological ordering.

g) T F Given a graph G = (V, E) with positive edge weights, the Bellman-Ford algorithm and Dijkstra's algorithm can produce different shortest paths despite always producing the same shortest-path weights.

h) T F Dijkstra's algorithm may not terminate if the graph contains one negative-weight edge.

i) T F The height of any binary search tree with n nodes is O(log n).

j) T F Knapsack problem is not an NP-Complete Problem because it can be efficiently solved using dynamic programming technique.

4. Use perfect hashing to store the set of $K = \{10, 40, 60, 75\}$, with its outer hash function of $h(k) = ((a \cdot k + b) \bmod p) \bmod m$, where $a = 3$, $b = 42$, $p = 101$, and m (i.e., the outer hash table size) = 9. Illustrate the perfect hashing result under K.

5. For a given B-tree of height h and with the minimum node degree of $t \geq 2$, what is the maximum number of keys held in such a B-tree?

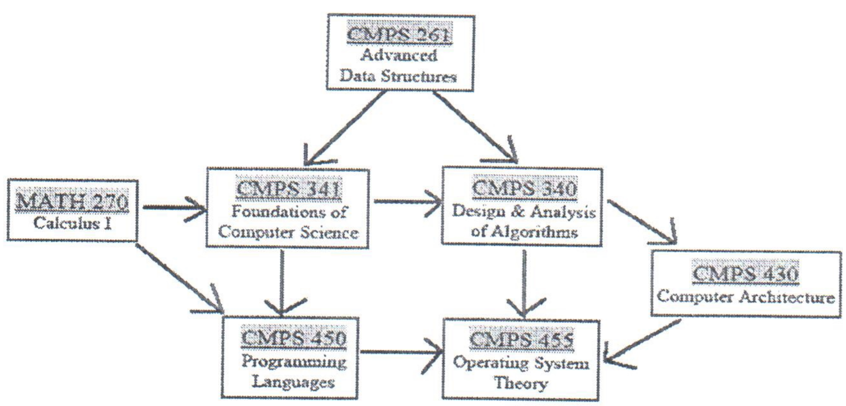
6. Show your construction of an optimal Huffman code for the set of 10 frequencies: **a**:2 **b**:6 **c**:5 **d**:8 **e**:13 **f**:21 **g**:34 **h**:19 **i**:27 **j**:9 **k**:15 **l**: 43.

LONG QUESTIONS (Answer all four questions, each carrying 15 points.)

1. Knapsack problem: Given a knapsack with weight constraint W and n items. Each item i has value $v[i]$ and weight $w[i]$. Find the maximal value the knapsack can take.
   The time complexity of a recursive solution as shown below, is exponential. Please provide **a top-down dynamic programing solution**. (Hint: do not rewrite the code! Add just a few lines of code to the following solution.)

```
M(n, W)
{
        if (n == 0 or W == 0) return 0;

        if (w[i] > W)

                result = M(n-1, W);

        else

                result = max{v[i] + M(n-1, W - w[i]), M(n-1, W)};

        return result;
}
```

2. The following graph shows a sequence of courses with their dependencies on other courses. A directed edge from course a to course b indicates that course a must be taken before course b can start. Please apply topological sort to find an ordering of these courses that conforms to the given dependencies. (Show each step.)



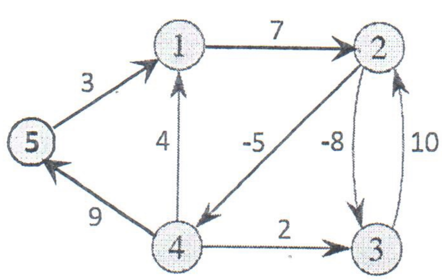
3. An optimal binary search tree (OBST) for a given set of keys with known access probabilities ensures the minimum expected search cost for key accesses. Given the set of four keys with their access probabilities of $k_1 = 0.23$, $k_2 = 0.15$, $k_3 = 0.1$, $k_4 = 0.2$, respectively, and five non-existing probabilities of $d_0 = 0.1$, $d_1 = 0.05$, $d_2 = 0.06$, $d_3 = 0.03$, $d_4 = 0.08$, (a) construct OBST following dynamic programming with memoization for the given four keys and (b) demonstrate the constructed OBST, which contains all four keys (k_1, k_2, k_3, k_4) and five non-existing dummies (d_0, d_1, d_2, d_3, d_4).

(Show your work using the three tables, for expected costs: $e[i, j]$, access weights: $w[i, j]$, and $root[i, j]$, with i in $e[i, j]$ and $w[i, j]$ ranging from 1 to 5, j in $e[i, j]$ and $w[i, j]$ ranging from 0 to 4, and both i and j in $root[i, j]$ ranging from 1 to 4.)

```
OPTIMAL-BST(p, q, n)
let e[1..n+1, 0..n], w[1..n+1, 0..n],
        and root[1..n, 1..n] be new tables
for i = 1 to n + 1
    e[i, i-1] = q_{i-1}
    w[i, i-1] = q_{i-1}
for l = 1 to n
    for i = 1 to n - l + 1
        j = i + l - 1
        e[i, j] = ∞
        w[i, j] = w[i, j-1] + p_j + q_j
        for r = i to j
            t = e[i, r-1] + e[r+1, j] + w[i, j]
            if t < e[i, j]
                e[i, j] = t
                root[i, j] = r
return e and root
```

4. The Floyd-Warshall algorithm (FW) obtains all pairs of shortest paths in a weighted directed graph with n nodes. Consider the graph given below. What is the general recursive equation of $d_{i,j}^{(k)}$ for the shortest-path weight of any path between i and j with an intermediate vertice k where k belongs to $\{1, 2, 3, \ldots, n\}$? $d_{ij}^{k} = \begin{cases} w_{ij} & \text{if } k=0 \\ \min(d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1}) & \text{if } k>0 \end{cases}$

Derive all distance matrices $D^{(k)}$ following FW so that the $d_{i,j}^{(n)}$ element of final matrix $D^{(n)}$ denotes $\delta(i, j)$ for every vertex pair $<i, j>$ for all $i, j \in \{1, 2, 3, 4, 5\}$.



Good Luck!