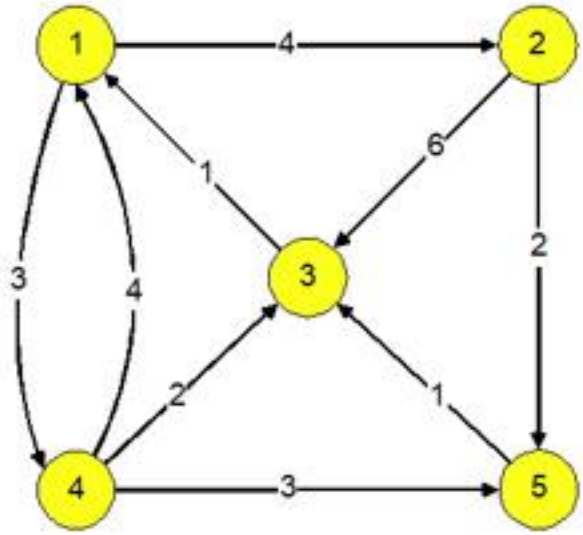
Question 1

Correct 20.00 points out of 20.00 Flag question

1Trace the action of Dijkstra's shortest path algorithm in generating the single-source shortest path tree rooted at vertex **2** by completing the distance and parent arrays as successive vertices are added to the shortest path tree. Use **99999** for ∞ and **-1,** for **null** if a vertex has no parent when a vertex is added to the tree. [15 points]



Distance and Parent Arrays before Adding the Root Vertex

Vertex	1	2	3	4	5
Distance	99999 ✓	0 ✓	99999 ✓	99999 ✓	99999 ✓
Parent	-1 ✓	-1 ✓	-1 ✓	-1 ✓	-1 ✓

Distance and Parent Arrays before Adding the Root Vertex

Vertex	1	2	3	4	5
Distance	99999 ✓	0 ✓	6 ✓	99999 ✓	2 ✓
Parent	-1 ✓	-1 ✓	2 ✓	-1 ✓	2 ✓

Distance and Parent Arrays before Adding the Root Vertex

Vertex	1	2	3	4	5
Distance	99999 ✓	0 ✓	3 ✓	99999 ✓	2 ✓
Parent	-1 ✓	-1 ✓	5 ✓	-1 ✓	2 ✓

Distance and Parent Arrays before Adding the Root Vertex

Vertex	1	2	3	4	5
Distance	4 ✓	0 ✓	3 ✓	99999 ✓	2 ✓
Parent	3 ✓	-1 ✓	5 ✓	-1 ✓	2 ✓

Distance and Parent Arrays before Adding the Root Vertex

Vertex	1	2	3	4	5
Distance	4 ✓	0 ✓	3 ✓	7 ✓	2 ✓
Parent	3 ✓	-1 ✓	5 ✓	1 ✓	2 ✓

Give $\pi(2, 4)$, the full shortest path from vertex **2** to vertex **4**, as a sequence of vertices with each vertex, except for the last, followed by a comma and a space, beginning with **2** and ending with **4**.

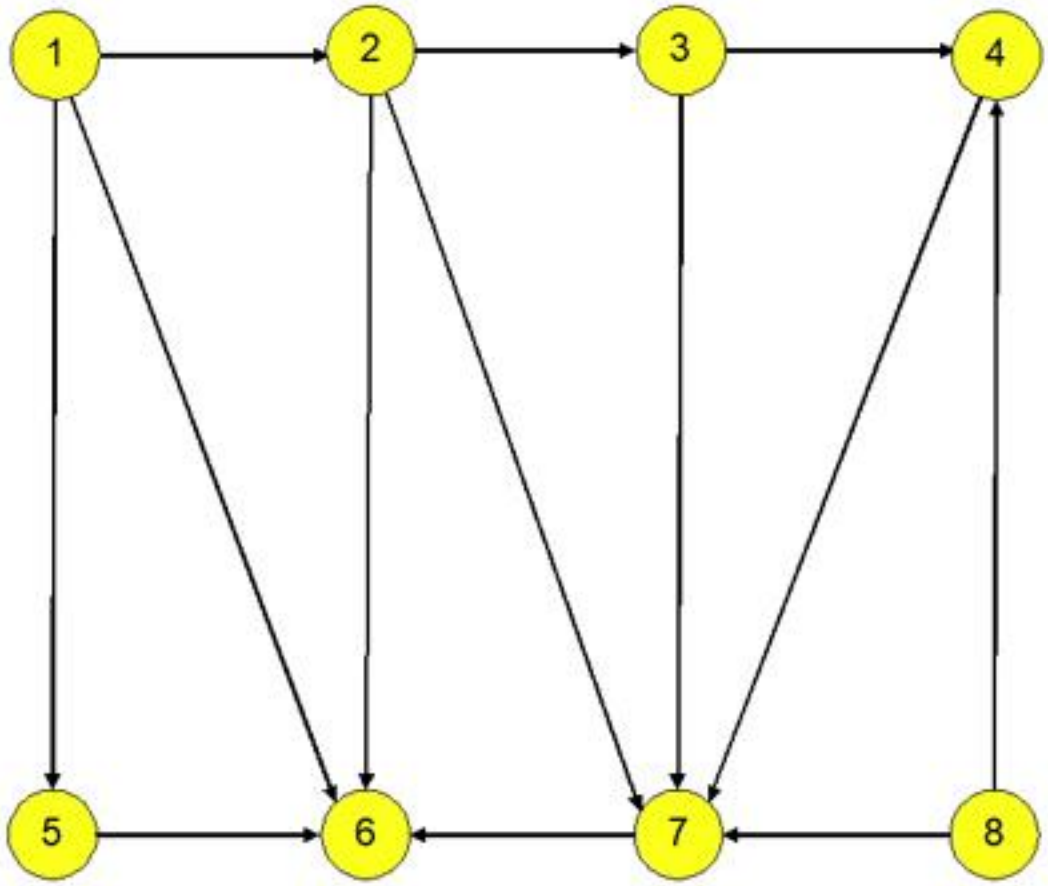
2, 5, 3, 1, 4 ✓ [3 points]

What is $\delta(2, 4)$, the shortest distance from vertex **2** to vertex **4**? 7 ✓ [2 points]

Question 2

Correct 20.00 points out of 20.00 Flag question

Give the sequence of vertices that represent a valid topological ordering of the vertices of the graph show below using the indicated strategy. During the execution of each strategy, whenever two or more vertices are valid choices for the next vertex to be chosen resolve the tie based on lexicographical ordering of the vertices.



(a) Generate a topological ordering by choosing successive vertices based on the in-degree strategy. [10 points]

Topological Listing (In-Degree-Based)							
1 ✓	2 ✓	3 ✓	5 ✓	8 ✓	4 ✓	7 ✓	6 ✓

(b) Generate a topological ordering by applying the reversed post-order depth-first-search-based strategy. [10 points]

Topological Listing (In-Degree-Based)							
8 ✓	1 ✓	5 ✓	2 ✓	3 ✓	4 ✓	7 ✓	6 ✓

Question 3

Correct 20.00 points out of 20.00 Flag question

Consider a closed hash table (open addressing) H[0:10] that holds at most TABLE_SIZE = 11 entries for exercises (a)and (b). Provide a numeric value for the probability in (c).

(a) Show the hash table when 33, 5, 12, 19, 10, 38, 32, 7 and 39 are mapped into an initially empty hash table in the order listed using the hash function *h(k) = k mod 11* and resolving collision by quadratic probing. Use H[i] = -1 to indicate that no entry is stored at address i in the table after all the keys are inserted. [11 points]

Hash Table (H[0:10])										
33 ✓	12 ✓	-1 ✓	32 ✓	39 ✓	5 ✓	38 ✓	7 ✓	19 ✓	-1 ✓	10 ✓

(b) What is the total number of collisions? 6 ✓ Which key gives the maximum number of collisions? 39 ✓ [4 points]

(c) Give the probability that at least one collision occurs when three randomly generated keys are hashed into an initially empty hash table of size 10? 0.28 ✓ [5 points]

Question 4

Complete Points out of 20.00 Flag question

Let Σ = {0, 1} be the alphabet from which strings are drawn. P = 1010 and H = 1001011010 are strings in Σ^*, the set of all strings constituted by combining symbols in Σ. In this case, Σ^* denotes the set of all bit strings.

(a). Compute the longest-prefix-proper suffix function for P, denoted $\pi[P]$. Copy and paste the chart below representing $\pi[P]$ in your response area and then complete it. [5 points]

$\pi[P]$ for P=1010

(b). Trace the action of the Knuth-Morris-Pratt (KMP) string matching algorithm in finding the first occurrence of *P* in *H* by completing the chart below. The second row of the chart contains the host string. The third row of the table contains characters of the *P* up to the first mismatch characters of P and H. The comparisons are made prior to the first shift of P. The x in the first row, above the column, indicates that pattern and host characters did not match. Complete the chart by showing all the shifts of the pattern, beginning with the first shift on the third row. Include characters of the pattern up to and including the mismatch character. Put an *x* on the first row for the column heading of the mismatch characters if one is not already there. Use as many rows as needed with the last row that you complete representing a valid shift, the shift at which a match of the pattern is found in the host. Copy and paste the chart below in your response area and complete it. [10 points]

A Hand-trace of KMP Matcher in Finding P=1010 in H=1001011010

		x							
1	0	0	1	0	1	1	0	1	0
1	0	1							

c. How many pairs of characters of the pattern and host were compared during the execution of the algorithm? How many pairs of characters of the pattern and the host would the brute-force algorithm have compared? [5 points]

A.

B.

A Hand-trace of KMP Matcher in Finding P=1010 in H=1001011010

		x				x			
1	0	0	1	0	1	1	0	1	0
1	0	1							
		1							
			1	0	1	0			
					1	0			
						1	0	1	0

C. 13; 16

Question 5

Complete Points out of 20.00 Flag question

In (a)-(c), assume that the nodes of the b-trees are enumerated in level order and give the node-parent tabular representation of each tree. Each node must be represented by the sequence of characters that it contains and so must its parent node.

(a) Insert the following entries, in the order listed, into an initially empty B-tree of order 3.

j e s i r x c l n t u

i. Give the tree after the insertion of **c**, the seventh key. [5 points]

ii. Give the tree after the insertion of **u**, the last key. [5 points]

(b) Assuming in-order predecessor replacement strategy, give the tree when **t** is deleted from the tree after all the keys are inserted. [5 points]

(c) Give all B-trees of order 5 that can be constructed each of which contains the keys **A**, **B**, **C**, **D**, **E**, **F**, **G** and **H**. [5 points]

(a)(i)

Node Parent

[j] NULL

[e] [j]

[s] [j]

[c] [e]

[i] [e]

[r] [s]

[x] [s]

(a)(ii)

Node Parent

[js] NULL

[e] [js]

[n] [js]

[u] [js]

[c] [e]

[i] [e]

[l] [n]

[r] [n]

[t] [u]

[x] [u]

(b)

Node Parent

[j] NULL

[e] [j]

[ns] [j]

[c] [e]

[i] [e]

[l] [ns]

[r] [ns]

[u] [ns]

[x] [ns]

(c)

1. **Node Parent**

[CF] NULL

[AB] [CF]

[DE] [CF]

[GH] [CF]

2. **Node Parent**

[D] NULL

[ABC] [D]

[EFGH] [D]

3. **Node Parent**

[E] NULL

[ABCD] [E]

[FGH] [E]