KLE Technological University
Creating Value, Leveraging Knowledge

Earlier known as
B. V. B. College of Engineering & Technology

Department of Computer Science and Engineering

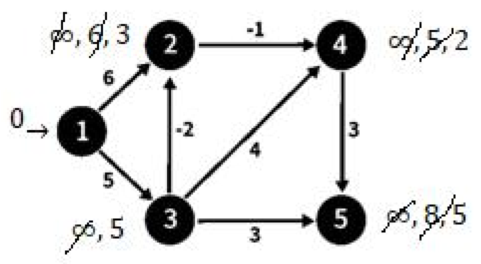
ISA EXAM II

Scheme and solutions

Course : Design and Analysis of Algorithms	
Course Code : 24ECSC205	Semester : III
Date of Exam : 19/12/2025	Duration : 75 mins
Note: (i) Answer any two full questions. (ii) Each full question carries equal marks.	

Q.No.	**Solutions**	**Marks**

1.a	Write an algorithm for Insertion sort
	ALGORITHM InsertionSort(A[0..n-1])
	// Sorts a given array using insertion sort
	// Input: An array A[0..n-1] of orderable
	// elements
	// Output: Array A[0...n-1] sorted in
	// ascending order
	for i← 1 to n -1 do
	v ←A[i]
	j ← i - 1
	while j >= 0 and A[j] > v do
	A[j + 1] ← A[j]
	j ← j -1
	A[j + 1] ← v
		4

1b	Apply Bellman-Ford on the following graph. Vertices: 1,2,3,4,5 Edges with weights in following order: (1→2,6), (1→3,5), (2→4,−1), (3→2,−2), (3→4,4), (3→5,3), (4→5,3).  ~~∞~~, ~~6~~, 3 ~~∞~~, ~~5~~, 2 0→ ~~∞~~, 5 ~~∞~~, ~~8~~, 5 1 2 3 4 5 6 -1 -2 4 3 5 3	6
1c	Rabin Karp TEXT: CDEOBAZZINGA PATTERN: ZINGA	10 (3+3+4)

Hash	A	B	C	D	E	G	I	O	N	Z

ISA EXAM II

Scheme and solutions

Values	0	1	2	3	4	5	6	7	8	9

Pattern	Z	I	N	G	A
	9	6	8	5	0

Hash Code for Pattern: $9 \times 10^4 + 6 \times 10^3 + 8 \times 10^2 + 5 \times 10^1 + 0 \times 10^0 = 96850$

Define Rolling hash and

Compare Efficiency over Brute Force algorithm

2a — Efficiency of Merge Sort. — 4

T(n) = 2T(n/2) + n gives us O(nlogn)
T(n)=2T(n/2)+n
T(n) =2(2T(n/4) +cn/2) +cn
  = 4T(n/4)+2cn
  = 4(2T(n/8)+cn/4)+2cn
  * *
  = 2k T(1)+kCn.
  = an + cn log n
  if sk <n<=2k+1, then T(n)<=T(2k+1)
  T(n)=O(n log n)

2b — 6 (3+3)

let us consider searching for the pattern BAOBAB.
The bad-symbol table looks as follows:

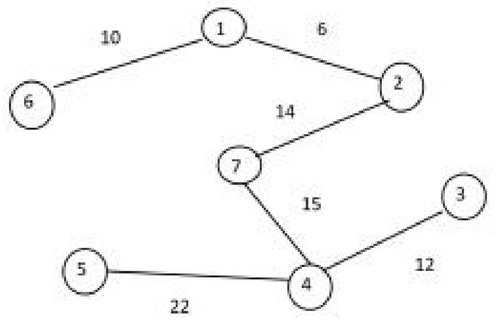
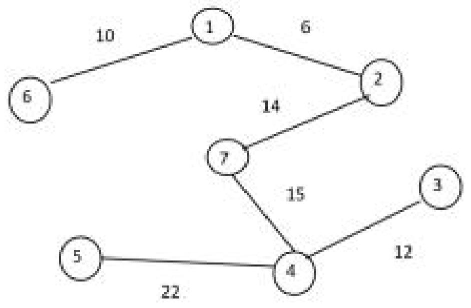
c	A	B	O	*
t1(c)	1	2	3	6

The good-suffix table is filled as follows:

k	pattern	d2
1	$\text{BAOBA}\underline{\text{B}}$	2
2	$\text{BAOB}\underline{\text{AB}}$	5
3	$\text{BAO}\underline{\text{BAB}}$	5
4	$\text{BA}\underline{\text{OBAB}}$	5
5	$\text{B}\underline{\text{AOBAB}}$	5

ISA EXAM II

Scheme and solutions

2c	Prim's and Kruskal's algorithms may compute different minimum spanning trees when run on the same graph, is this true or false? Ans: **True** 2M The following graph gives the same (output tree) total cost. Efficiency class of both the algorithm: O(n 2) **2Marks**  10 1 6 6 2 14 7 15 3 5 4 12 22 10 1 6 6 2 14 7 15 3 5 4 12 22 Prim's algorithm: cost 79 **(3 Marks)** Kruskal's Algorithm: cost 79 **(3 Marks)**	10 (2+2+3+3)

3a	Dijkstra's Algorithm
	ALGORITHM Dijkstra(G, s)
	// Dijkstra's algorithm for single source shortest path
	// Input: A weighted connected graph G(V, E) with non-negative weights and its vertex s
	// Output: the length dv of a shortest path from s to v and its penultimate vertex pv for every vertex v in V
	Initialize(Q) // Initialize vertex priority queue to empty
	for every vertex v in V do
	dv ∞
		4

ISA EXAM II

Scheme and solutions

```
pv      null
Insert (Q, v, dv) // Initialize vertex priority in priority queue
ds      0
Decrease (Q, s, ds) // Update priority of s with ds
VT      Ø
for i      0 to |V| - 1 do
u*      DeleteMin(Q)
VT = VT U {u*}
for every vertex u in V – VT that is adjacent to u* do
if du* + w(u*, u) < du
du      du* + w(u*, u)
pu      u*
Decrease (Q, u, du)
```

3b — Explain KMP with following text and pattern — 6 (3+3)

Text: ABABCDABCDABDE   Pattern: ABCDABD

Π table	A	B	C	D	A	B	D
LPS	0	0	0	0	1	2	0

A	B	A	B	C	D	A	B	C	D	A	B	D	E
A	**B**	C	D	A	B	D							
		A	**B**	**C**	**D**	**A**	**B**	D					
						A	**B**	**C**	**D**	**A**	**B**	**D**	

3 c — Floyd's-algorithm (4 Marks) — 10 Marks (4+6)

ALGORITHM Floyd (W[1..n,1..n])

KLE Technological University | Creating Value, Leveraging Knowledge

Earlier known as
B. V. B. College of Engineering & Technology

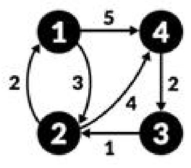
Department of Computer Science and Engineering

ISA EXAM II

Scheme and solutions

```
// Implements Floyd's algorithm for all pair shortest path problem
// Input: The weight matrix W of the graph with no negative length cycle
// Output: The distance matrix of the shortest path's lengths
D      W
for k      1 to n do
for i      1 to n do
for j      1 to n do
D [i, j]      min {D[i, j], D[i, k] +D[k, j]}
return D
```

$D^1[i][j]=\min(D^0[i][j],D^0[i][k]+dist[k][j])$



$$D^0=\begin{bmatrix}0 & 3 & \infty & 5\\ 2 & 0 & \infty & 4\\ \infty & 1 & 0 & \infty\\ \infty & \infty & 2 & 0\end{bmatrix}\quad D^1=\begin{bmatrix}0 & 3 & \infty & 5\\ 2 & 0 & \infty & 4\\ \infty & 1 & 0 & \infty\\ \infty & \infty & 2 & 0\end{bmatrix}$$

$$D^2=\begin{bmatrix}0 & 3 & \infty & 5\\ 2 & 0 & \infty & 4\\ 3 & 1 & 0 & \infty\\ \infty & \infty & 2 & 0\end{bmatrix}\quad D^3=\begin{bmatrix}0 & 3 & \infty & 5\\ 2 & 0 & \infty & 4\\ 3 & 1 & 0 & 5\\ 5 & 3 & 2 & 0\end{bmatrix}\quad D^4=\begin{bmatrix}0 & 3 & 7 & 5\\ 2 & 0 & 6 & 4\\ 3 & 1 & 0 & 5\\ 5 & 3 & 2 & 0\end{bmatrix}$$

(6 Marks)