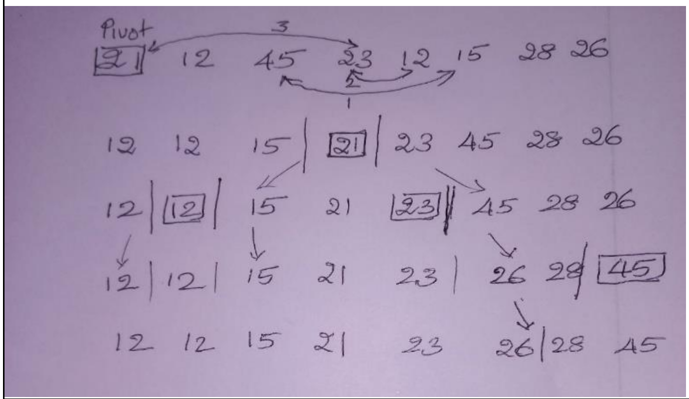
Internal Assessment Test 2 – Aug 2023 ANSWER KEY

Sub:	**Design and Analysis of Algorithms**				Sub Code:	**21CS42**	Branch:	**CSE**
Date:	**10/08/2023**	Duration:	90 mins	Max Marks:	50	Sem/Sec:	IV/A,B&C	OBE

Answer any FIVE FULL Questions

		MARKS	CO	RBT
1	a. Apply quicksort on the array with keys [21, 12, 45, 23, 12, 15, 28, 26]. Show the array contents at each level of recursive calls.	6	CO2	2



		MARKS	CO	RBT
	b. A modified version of the quicksort is given below.	4	CO2	3

```
Algorithm NewQuicksort(A[], l, r)
//Inputs: Array A, l & r – start and end indices index of the sub-array to be sorted.
If (l<r)
    p = newpartition(A, l, r)
    Quicksort(A[], l, p-1)
    Quicksort(A[], p+1, r)
```

The new partitioning function identifies a pivot element that is the median of the elements from index l to r and partitions the array around this pivot. In otherwords, p will be equal to (l+r)/2. The newpartition function has a time complexity of Θ(n). Write the recurrence equation for the NewQuicksort algorithm and solve it using Master's method.

Recurrence equation: $T(n) = 2T(n/2) + \Theta(n)$

Master theorem can be applied for reccurent relations of the form
$T(n) = aT(n/b) + f(n)$, where $f(n) \in \Theta(n^k)$, $k>=0$

In this case, a=2, b=2 and k=1 and $b^k=2$. Second rule of master method is applicable as $a= b^k$. Applying second rule, we get $T(n) = \Theta(n \log n)$

1 mark for recurrence equation and 3 for solving it

		MARKS	CO	RBT
2	Explain divide and conquer technique. Write a recursive algorithm for finding maximum and minimum element from a list. Compare the same with straight forward approach in terms of number of comparisons made.	3+5+2	CO2	2

Divide and conquer approach solves a problem using to 3 steps:

1. Divide instance of problem into two or more smaller instances (In most cases, sub problems are of the same type as original problem)
2. Solve smaller instances recursively

3. Obtain solution to original (larger) instance by combining these solutions

Divide and conquer recursive algorithm for finding min and max.

```
Algorithm Recminmax(a[], i, j, max, min)
Input: array a, start index i, last index j
Output: min and max values
if (i==j)
        max=a[i]; min = a[i];
else if (i==j-1)
        if a[i] < a[j]
                max=a[j]; min = a[i];
        else
                max=a[i]; min = a[j];
else
        mid = (i+j) /2;
        Recminmax(a, i, mid, max, min);
        Recminmax(a, mid+1, j, max1, min1);
        if (max1 > max) max = max1
        if (min1 < min) min = min1
```

Number of comparisons for straight forward approach: 2(n-1)
Number of comparisons for recursive approach: 3n/2 - 2

3	a. Explain decrease and conquer method	3	CO2	1

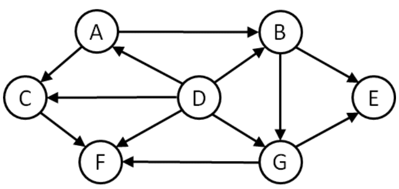
In decrease and conquer method, size of the problem is reduced in each iteration. When the size becomes small enough, problem is solved using a straight forward methos. There are three variations in decrease and conquer

- Decrease by a constant. Example Exponential using $a^n = a * a^{n-1}$
- Decrease by a constant factor: Example

$$a^n = \begin{cases} (a^{n/2})^2 & \text{if } n \text{ is even and positive,} \\ (a^{(n-1)/2})^2 \cdot a & \text{if } n \text{ is odd,} \\ 1 & \text{if } n = 0. \end{cases}$$

- Variable size decrease: Example $\gcd(m, n) = \gcd(n, m \bmod n).$

	b. Illustrate the topological sorting algorithm on the graph below using DFS method.	7	CO2	3



If we start DFS from a, the nodes have the following (visited/popped out) order

A: (1/6), B (2/4), E(3/1), G(4/3), F(5/2), C(6/5), D(7/7)

Listing the nodes in the reverse order of being popped out, we get the topological order as D, A, C, B, G, F, E

4	a. What is the best case, worst case and average case time complexity of insertion sort?	[3]	CO2	L2

Best Case: O(n) when the array is already sorted

Average Case: $O(n^2)$
Worst Case: $O(n^2)$

b. Solve the greedy knapsack problem where sack capacity m=20, n=5, Profits=(30,42,25,18,15), Weights=(6,14,5,2,3) | [7] | CO 3 | L2

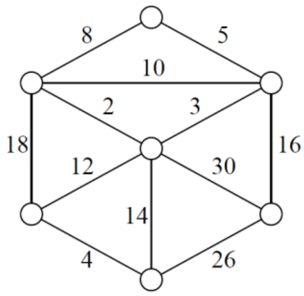
Calculate profit by weight for each object and take the items in decreasing order of profit/weight

Object #	1	2	3	4	5
Profit	30	42	25	18	15
Weight	6	14	5	2	3
Profit/Weight	5	3	5	9	5

Take all of item 4. Total profit: 0+18=18; Remaining Capacity 20-2=18
Take all of item 1. Total profit: 18+30=48; Remaining Capacity 18-6=12
Take all of item 3. Total profit: 48+25=73; Remaining Capacity 12-5=7
Take all of item 5. Total profit: 73+15=88; Remaining Capacity 7-3=4
Take 4 units of item 2. Total profit: 88+3*4=100;
Remaining Capacity 4-4=0

Total profit is 100.

5 In a city, main places (shown as vertices in the graph) are located as shown in the diagram below. The distances between them are also shown in the diagram. The city corporation wants to lay water pipelines so that water reaches every location. However, they want to minimize the cost of laying pipelines. Use an appropriate algorithm to determine how the main places shown in diagram should be connected so that the total length of pipe required is a minimum. | [10] | CO3 | L3



Apply either Kruskal's or Prim's algorithm.

Applying Kruskal's algorithm

Sort the edges in increasing order of weights and add them to into MST as long as the new edge does not form a cycle. The order in which the edges considered are

1. 2: Add
2. 3: Add
3. 4: Add
4. 5: Add
5. 8: Discard (Forms cycle)
6. 10: Discard (Forms cycle)
7. 12: Add
8. 14: Discard (Forms cycle)
9. 16: Add

10. All others are discarded as they form cycles

Total cost of pipes: 2+3+4+5+12+16 = 42

6 a. What is Job sequencing with deadlines problem? For the given data, find the optimal job sequence and maximum profit using Greedy approach. [1+4] CO3 L3

Jobs	J1	J2	J3	J4	J5
Profits	10	12	5	23	7
Deadlines	2	2	3	2	1

Job Scheduling with deadline problem statement

We are given a set of n jobs. Associated with job i is an integer deadline di >= 0 and a profit pi> 0. For any job i, the profit pi is earned iff the job is completed by its deadline. To complete a job, one has to process the job on a machine for one unit of time (i.e. all jobs are 1 unit long). Find the maximum profit with optimal scheduling.

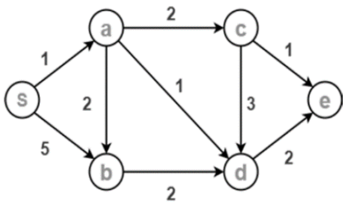
Solution to the given problem

Sort the Jobs in decreasing order and schedule them as late as possible.

1. J4 (Profit: 23): Schedule in slot 2
2. J2 (Profit: 12): Schedule in slot 1
3. J1 (Profit: 10): Discard as no free slot on or before 2
4. J5 (Profit: 7): Discard as no free slot on or before 1
5. J3 (Profit: 5): Schedule in slot 3.

Total profit: 23+12+5 = 40.

b. Apply Dijkstra's algorithm to find single source shortest path for the given graph by considering 'S' as the source vertex [5] CO3 L3



.

Vistied Nodes	Other nodes with Distance estimation
S(0)	**a(1)**, b(5), c(∞), d(∞),e(∞)
a(1)	b(5 1+2), c(1+2), **d(1+1)**,e(∞)
d(2)	**b(3)**, c(3), e(2+2)
b(3)	**c(3)**, e(4)
c(3)	**e(4)**
e(4)	