Solved Question Papers

USN

06CS62

Fourth Semester B.E. Degree Examination, Dec–Jan 2008–2009 Analysis and Design of Algorithms

Time: 3 hrs Max. Marks: 100

Note: 1. Answer any FIVE full questions, selecting at least TWO questions from each part A and part B.

PART A

1

a. Discuss the various stages of algorithm design and analysis process using flow chart. (10 Marks)

b. Explain important fundamental problem types of different categories. (10 Marks)

2

a. Explain in brief the basic asymptotic efficiency classes. (06 Marks)

b. Explain the method of comparing the order of the growth of two functions using limits. Compare order of growth of following functions (i) $\log_2 n$ and $\sqrt{n}$ (ii) $(\log_2 n)^2$ and $\log_2 n^2$. (09 Marks)

c. Discuss the general plan for analyzing efficiency of non-recursive algorithms. (05 Marks)

3

a. What is brute-force method? Explain sequential search algorithm with an example. Analyse its efficiency. (10 Marks)

b. Write the merge sort algorithm and discuss its efficiency. Sort the list E, X, A, M, P, L, E in alphabetical order using merge sort. (10 Marks)

4

a. What is divide–and–conquer technique? Apply this method to find multiplication of integers 2101 and 1130. (08 Marks)

b. Explain the differences between DFS and BFS. Solve topological sorting problem using DFS algorithm with an example. (12 Marks)

PART B

5

a. Explain bottom-up heap sort algorithm with an example. Analyse its efficiency. (10 Marks)

b. Write Horspool's algorithm. Apply Horspool algorithm to search for the pattern BAOBAB in the text BESS_KNEW_ABOUT_BAOBABA. (10Marks)

6

a. Write Warshall's algorithm. Apply Warshall's algorithm to find the transitive closure of the following Fig. No. 6(a) (10 Marks)

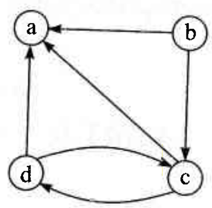


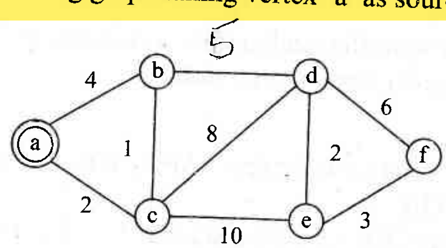
Fig. 6(a)

b. Solve the following Knapsack problem with given capacity W = 5 using dynamic programming. (10 Marks)

Item	*Weight*	*Value*
1	2	$12
2	1	$10
3	3	$20
4	2	$15

7

a. Write Dijkstra's algorithm and apply the same to find single source shortest paths problem for the following graph taking vertex 'a' as source in Fig. No. 7(a). (10 Marks)



b. What are decision trees? Explain the concept of decision trees for sorting algorithms with an example. (10 Marks)

8

a. Briefly explain the concepts of P, NP and NP complete problems. (10 Marks)

b. Explain back-tracking algorithm. Apply the same to solve the following instance of the subset–sum problem: $S = \{3, 5, 6, 7\}$ and $d = 15$. (10 Marks)

SOLUTIONS

Analysis and Design of Algorithms Solutions for Dec 2008–Jan 2009

PART A

1.

a. The stages of algorithm design and analysis process using flow chart.
Refer to Page No. 9–15 PAGE NOS OF ANANY LEVITIN'S BOOK SECOND EDITION

b. Important fundamental problem types
Refer to Page No. 17–21

2.

a. Basic asymptotic efficiency classes
Refer to Page No. 55

b. Using limits for comparing orders of growth
Refer to Page No. 53

c. Comparing

(i) $\log_2 n$ and $\sqrt{n}$

Refer Page No. 54, example 2.

(ii) $(\log_2 n)^2$ and $\log_2 n^2$

$$\lim_{x\to\infty} \frac{(\log_2 n)^2}{\log_2 n^2} = \frac{[(\log_2 n^2)]^1}{[\log_2 n^2]^1} = \frac{\frac{\log n}{n}}{\frac{c}{n}} = \log n$$

$$\lim_{x\to\infty} \log n = \infty$$

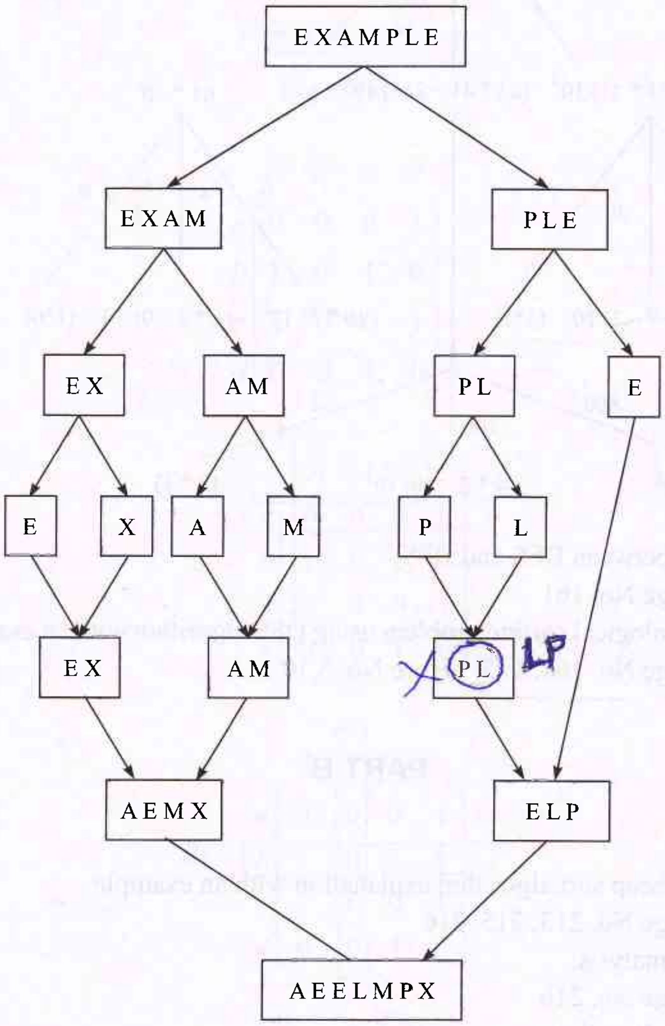
$(\log_2 n)^2$ has a larger order of growth than g(n)

c. General plan for analyzing efficiency of non-recursive algorithms.
Refer to Page No. 58–59

3.

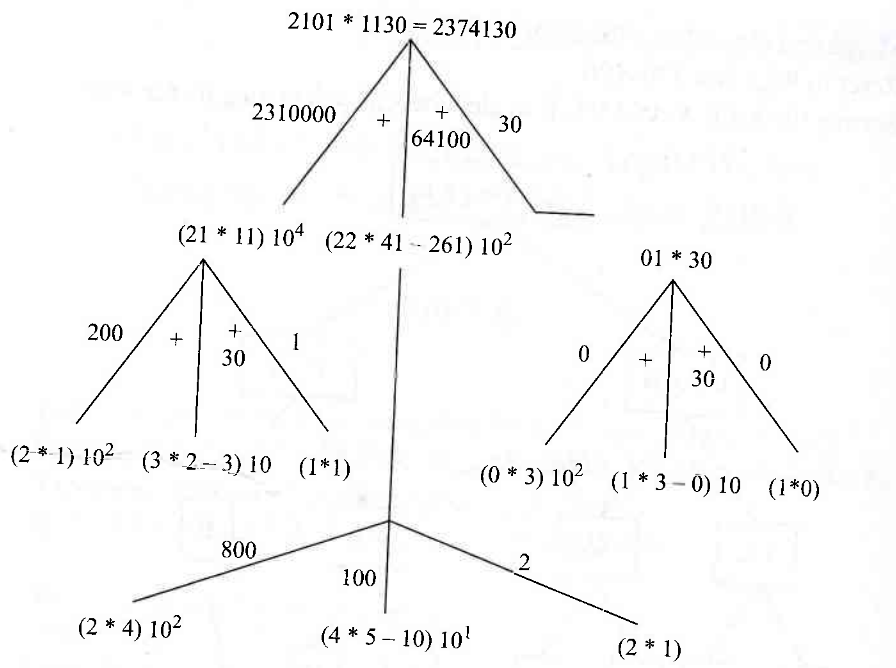
a. Brute-force method definition
Refer to Page No. 93/45
Sequential search algorithm with an example and its efficiency
Refer to Page No. 99
Efficiency analysis—Refer to Page No. 45–46

b. Merge sort algorithm, efficiency
Refer to Page No. 119–120
Sorting the list E X A M P L E in alphabetical order using merge sort



4.

a. Divide –and–conquer technique
Refer to Page No. 117
Divide–and–conquer to find multiplication of integers 2101 and 1130



b. Difference between DFS and BFS
   Refer to Page No. 161
   Solving topological sorting problem using DFS algorithm with an example
   Refer to Page No. 164, 165, Figure No. 5.10

PART B

5.
   a. Bottom-up heap sort algorithm explanation with an example.
      Refer to Page No. 213, 215–216
      Efficiency analysis:
      Refer to Page No. 216
   b. Horspool's algorithm
      Refer to Page No. 245/246.
      Pattern searching solution using Horspool's algorithm
      Refer to Answer 5 (c) of June/July 2009 Question Paper.

6.
   a. Warshall's algorithm
      Refer to Page No. 272
      Finding transitive closure of the given graph using warshall's algorithm.

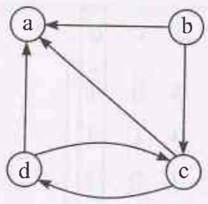


Fig. 6 (a)

$$\begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 0 & 0 & 1 \\ b & 1 & 0 & 1 & 0 \\ c & 1 & 0 & 0 & 1 \\ d & 0 & 0 & 1 & 0 \end{array}$$

$$R_0 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 0 & 0 & 1 \\ b & 1 & 0 & 1 & 0 \\ c & 1 & 0 & 0 & 1 \\ d & 0 & 0 & 1 & 0 \end{array}$$

$$R_1 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 0 & 0 & 1 \\ b & 1 & 0 & 1 & 1 \\ c & 1 & 0 & 0 & 1 \\ d & 0 & 0 & 1 & 0 \end{array}$$

$$R_2 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 0 & 0 & 1 \\ b & 1 & 0 & 1 & 1 \\ c & 1 & 0 & 0 & 1 \\ d & 0 & 0 & 1 & 0 \end{array}$$

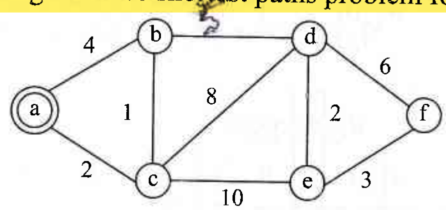
$$R_3 = \begin{array}{c|cccc|} & a & b & c & \boxed{d} \\ \hline a & 0 & 0 & 0 & 1 \\ b & 1 & 0 & 1 & 1 \\ c & 1 & 0 & 0 & 1 \\ \boxed{d} & 1 & 0 & 1 & 1 \end{array}$$

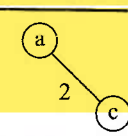
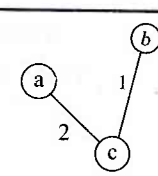
$$R_4 = \begin{array}{c|cccc|} & a & b & c & d \\ \hline a & 1 & 0 & 1 & 1 \\ b & 1 & 0 & 1 & 1 \\ c & 1 & 0 & 1 & 1 \\ d & 1 & 0 & 1 & 1 \end{array}$$

b. Knapsack problem solution using dynamic programming
Refer to Page No. 285, Example 1, Figure No. 8.13

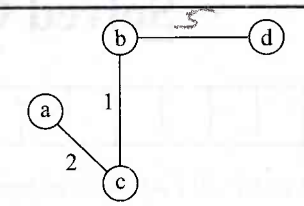
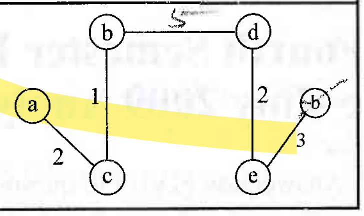
7.

a. Dijkstra's algorithm
Refer to Pages No. 307 and 309
Solving for single source shortest paths problem for the given graph.



Tree vertrics	*Remaining vertics*	*Illustration*
a(–, 0)	B(a, 4), c(a, 2), d (–, ∞), e(–, ∞), f(–, ∞)	
c(a, 2)	b(c, 2+1), d(c, 2+8), e(c, 2+10), f(–, ∞)	
b(c, 3)	d(b, 3+5), e(c, 12) f(–, ∞)	

Contd.

d(b, 8)	e(d, 8+2) f(d, 8+6)		
e(d, 10) f(e,13)	f(e, 10+3)		

Shortest path are

a to b: a ⟶ c ⟶ b of length 3

a to c: a ⟶ c of length 2

a to d: a ⟶ c ⟶ b ⟶ d of length 8

a to e: a ⟶ c ⟶ b ⟶ d ⟶ e of length 10

a to f: a ⟶ c ⟶ b ⟶ d ⟶ e ⟶ of length 13

b. Decision trees definition
Refer to Page No. 366
Decision trees for sorting algorithm
Refer to Page No. 367—368
Example
Refer to Pages No. 368/369, Figure No. 11.2/Figure No. 11.3

8

a. Concepts of P, NP and NP complete problems
Refer to Page No. 373—379

b. Back-tracking algorithm instance for subset sum problem using back-tracking.
Refer to Page No. 398, Figure No. 12.4

Solved Question Papers

USN

06CS62

Fourth Semester B.E. Degree Examination, June–July 2009 Analysis and Design of Algorithms

Time: 3 hrs Max. Marks: 100

Note: 1. Answer any FIVE full questions selecting at least TWO questions from each part.

PART A

1.

a. With figure, explain algorithm development process. (10 Marks)

b. Explain how priority queue can be implemented as unsorted array. (6 Marks)

c. Find GCD (60, 24) by applying Euclid's formula. Estimate the number of times computation is done in Euclid's method and in an algorithm based on checking consecutive integers from min (m, n) down to gcd (m, n). (4Marks)

2.

a. Explain all asymptotic notations used in algorithm analysis. (6 Marks)

b. Consider the following algorithm

```
Algorithm Enigma (A[0.. n-1, 0.. n-1])
    for i → 0 to n-2 do
      for j ← i+1 to n-1 do
        if A[i, j] ≠ A[j, i]
          return false
      end for
    end for
        return true
end algorithm
```

i. What does this algorithm compute?

ii. What is its basic operation?

iii. How many times is the basic operation executed?

iv. What is the efficiency class of this algorithm?

v. Can this algorithm be further imported? (10 Marks)

Consider the following recursive algorithm for computing the sum of the first n cubes.

$S(n) = 1^3 + 2^3 + 3^3 + \dots + n^3$

```
Algorithm S (n)
   if (n = 1) return 1
     else return (S (n-1) + n* n* n)
end algorithm
```

Set up and solve a recurrence relation for the number of times the basic operation of the algorithm is executed. (04 Marks)

3.

a. Write the quick sort algorithm. Trace the same on data set 5, 3, 1, 9, 8, 2, 4, 7. (10 Marks)

b. Write an algorithm to find the height of binary tree. (4 Marks)

c. Outline an exhaustive search algorithm to solve a travelling salesman problem. (6 Marks)

4.

a. Consider a set of 13 elements in an array list. State the elements of array that require the largest number of key comparisons when searched for by binary search. Find the average number of key comparisons made by search in successful search and unsuccessful search in this array. (6 Marks)

b. Write depth first search algorithm. (8 Marks)

c. Briefly explain how breadth first search can be used to check connectness of a graph and also to find the number of components in a graph. (6 Marks)

PART B

5.

a. Design a presorting-based algorithm to find the distance between the 2 closest numbers in an array of 'n' numbers. Compare the efficiency of this algorithm with that of brute-force algorithm. (10 Marks)

b. Construct AVL tree for the set of elements 5, 6, 8, 3, 2, 4, 7. (6 Marks)

c. Apply Horspool's algorithm to search for the pattern BAOBAB in the text BESS NEW ABOUT BAOBABS

Also, find the total number of comparisons made. (4 Marks)

6.

a. For the input 30, 20, 56, 75, 31, 19, construct the open hash table. Find largest and average number of key comparisons in a successful search in the table. (6 Marks)

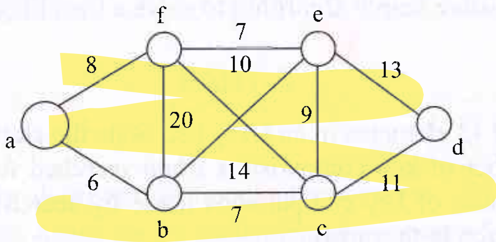
b. Explain dynamic programming. (4 Marks)

c. Write the formula to find the shortest path using Floyd's approach. Use Floyd's method to solve the following all pairs shortest paths problem. (10 Marks)

$$\begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & \infty & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \infty & 0 \end{bmatrix}$$

7.

a. Use Kruskal's method to find min cost spanning tree for the following graph. (6 Marks)



b. Write Huffman tree construction algorithm. (8 Marks)

c. Draw the decision tree for the 3-elements insertion sort. (6 Marks)

8.

a. Differentiate between back tracking and branch-and-bound algorithm. (6 Marks)

b. Draw the state space tree to generate first solution to 4-queens problem. With the first solution, generate another solution, making use of board's symmetry. (8 Marks)

c. Explain P and NP problems. (6 Marks)

SOLUTIONS

Analysis and Design of Algoriths Solutions for June-July 2009

PART A

1.

a. Algorithm development process—Refer to Page No. 9–15, Figure No. 1.2.

b. Priority queue implementation as unsorted array—Refer to Page No. 26.

Priority queue can be implemented efficiently using heap data structure. Descending priority queue can be implemented using max heap. In a max heap largest element (element with highest priority) can always be found at index 1 of the array. Hence, the delete operation of the priority queue requires deletion of the element at index 1.

For example, consider the heap

0	1	2	3	4	5	6	7	
	10	9	8	5	6	7	3	

After delete operation, the heap looks like

0	1	2	3	4	5	6	
	9	6	8	5	3	7	

c. To find GCD (60, 24) by applying Euclid's formula—Refer to Page 4–5.

2.

a. Asymptotic notation used in algorithm analysis—Refer to Page No. 50–52.

b. For the given algorithm

i. algorithm checks if the given matrix is symmetric over the main diagonal.

ii. basic operation is comparison

iii. the basic operation is executed (in the worst case)

$$C_{worsst} = \sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1} 1$$

$$= \sum_{i=0}^{n-z} [(n-1) - (i+1) + 1]$$

$$= \sum_{i=0}^{n-z} (n-1-i)$$

$$= \sum_{i=0}^{n-2} (n-1) - \sum_{i=0}^{n-2} i$$

$$= (n-1) \sum_{i=0}^{n-z} 1 - \sum_{i=0}^{n-z} i$$

$$= (n-1)\,[(n-2) - 0 + 1] - \left[\frac{(n-2)(n-1)}{2}\right]$$

$$= (n-1)(n-1) - \frac{(n-2)(n-1)}{2}$$

$$= (n-1)\left[(n-1) - \frac{(n-2)}{2}\right]$$

$$= (n-1)\left[\frac{2n-2-n+2}{2}\right]$$

$$= (n-1)\left(\frac{n}{2}\right)$$

$$= \frac{n(n-1)}{2}\,(n^2)$$

$$= \frac{1}{2}\,n^2 \in (n^2)$$

iv. The efficiency class is quadratic.

v. The algorithm can be further improved.

For the given recursive algorithm to compute the sum of first *n* cubes, the basic operation is multiplication. Let M(n) be the number of times the basic operation is executed. Then

$$M(n) = \underbrace{M(n-1)}_{\text{to cumpute } S(n-1)} + \underbrace{1+1+1}_{\text{to multiply } S(n-1) \text{ by } n * n * n} \quad \text{for } n > 1$$

$M(1) = 0$

Solving the relation, we have

$$\begin{aligned} M(n) &= M(n-1)+3 \\ &= [M(n-2)]+3+3 \\ &= M(n-2)+3+3 \\ &= [M(n-3)+3]+3+3 \end{aligned}$$

$$= \sum_{i=0}^{n-z} (n-1-i)$$

$$= \sum_{i=0}^{n-2} (n-1) - \sum_{i=0}^{n-2} i$$

$$= (n-1) \sum_{i=0}^{n-z} 1 - \sum_{i=0}^{n-z} i$$

$$= (n-1)\,[(n-2) - 0 + 1] - \left[\frac{(n-2)(n-1)}{2}\right]$$

$$= (n-1)(n-1) - \frac{(n-2)(n-1)}{2}$$

$$= (n-1)\left[(n-1) - \frac{(n-2)}{2}\right]$$

$$= (n-1)\left[\frac{2n-2-n+2}{2}\right]$$

$$= (n-1)\left(\frac{n}{2}\right)$$

$$= \frac{n(n-1)}{2}\,(n^2)$$

$$= \frac{1}{2}\,n^2 \in (n^2)$$

iv. The efficiency class is quadratic.

v. The algorithm can be further improved.

For the given recursive algorithm to compute the sum of first *n* cubes, the basic operation is multiplication. Let M(n) be the number of times the basic operation is executed. Then

$M(n) = M(n-1)$ to cumpute $S(n-1)$	$+1+1+1$ to multiply $S(n-1)$ by $n * n * n$	for $n > 1$

$M(1) = 0$

Solving the relation, we have

$M(n) = M(n-1)+3$

$= [M(n-2)]+3+3$

$= M(n-2)+3+3$

$= [M(n-3)+3]+3+3$

= M
= M
= M
when I =
=
=
=
M(n) =
∈

3.
a. Quick sor
Quick sor
b. Algorithm
c. Exhaustiv

4.
a. In a set of
compariso
i. key n
ii. some
The averag
i. Suc
C
ii. Uns
C

b. DFS algorit
c. Application

5.
a. Presorting-ba
numbers in a
brute-force al

= M(n – 3)+3+3+3
= M(n – 3)+3*3
= M(n – i)+3*i
when I = n – 1
= M(n–(n–1))+3*(n–1)
= M(1)+3*(n–1)
= 0+3*(n–1)
M(n) = 3n–3
∈ 0(n)

3. \

a. Quick sort algorithm—Refer to Page No. 124—125.
Quick sort to sort 5, 3, 1, 9, 8, 2, 4, 7—Refer to Page No. 126, Figure 4.3.

b. Algorithm to find the height of binary tree—Refer to Page No. 133.

c. Exhaustive search algorithm to solve TSP—Refer to Page No. 109.

4.

a. In a set of 13 elements, the elements that require the largest number of key comparison are
   i. key not present in the list
   ii. some cases of successful searches

The average number of key comparisons made by binary search in

i. Successful searchk

$C_{avg}(n) \cong \log_2 (n - 1)$
$= \log_2 (13 - 1)$
$= 4 - 1$
$= 3$

ii. Unsuccessful search

$C_{avg}(n) \cong \log_2 (n + 1)$
$= \log_2 (13 + 1)$
$= 4$

b. DFS algorithm—Refer to Page No. 157–158.

c. Application of BFS—Refer to Page No. 159–160.

PART B

5.

a. Presorting-based algorithm to find the distance between the 2-closest numbers in an array of 'n' numbers and its efficiency comparison with brute-force algorithm

ALGORITHM Present Minimum Distance (A[0….n–1])
//Solves the problem of finding minimum distance
//between the 2 closest elements
//Input : An array A[0….n–1] of orderable elements
//Output: Returns the minimum distance
Sort the array

```
Mindist = A[1] - A[0]
for i ← 2 to n-1
  if (|A[i] - A[i - 1]|)
    mindist = |A[i] - A[i - 1]|
return mindist
```

worst cast efficiency is

$$T(n) = T_{sort}(n) + T_{scan}(n)$$
$$\in \theta(n\log n) + \theta(n)$$
$$\in \theta(n\log n)$$

The efficiency of brute-force algorithm is $\theta(n^2)$.

This is because there will be $\frac{n(n-1)}{2}$ number of comparisons.

This clearly shows that preorder-based algorithm is more efficient compared to the brute-force algorithm to find the minimum distance.

b. All tree construction for the set of elements 5, 6, 8, 3, 2, 4, 7—Refer to Page No. 207.

c. Horspool's algorithm to search the pattern BAOBAB in the given text
Shift table

Chracter (c)	A	B	C	…	O	…	Z	-
Shift t(c)	1	2	6	6	3	6	6	6

B	E	S	S	-	K	N	E	W	-	A	B	O	U	T	-	B	A	O	B	A	B	S
B	A	O	B	A	B																	
						B	A	O	B	A	B											
																B	A	O	B	A	B	

Number of character comparisons = ~~11~~ ~~12~~ 13 No. of steps = 5 SHIFTS = 4

6.

a. To construct open hash table for the input 30, 20, 56, 76, 31, 19
Assumption: Let the hash function be
h(k) = (sum of digits of k)mod 10
h(30) =(3+0) mod 10 = 3

h(20) =(2+0) mod 10 = 2
h(56) =(5+6) mod 10 = 1
h(75) =(7+5) mod 10 = 2
h(31) =(3+1) mod 10 = 4
h(19) =(1+9) mod 10 = 0

Keys	30	20	56	75	31	19
hash address	3	2	1	2	4	0

0	1	2	3	4	5	6	7	8	9
↓	↓	↓	↓	↓					
19	56	20	30	31					
		↓							
		75							

The load factor = α = n/m

$$= \frac{\text{Total number of keys}}{\text{Total number of cells}}$$

$$= \frac{6}{10} = 0.6 = 60\%$$

Key comparisons in successful searches is

$$S \cong 1 + \frac{\propto}{2}$$

$$= 1 + \frac{0.6}{2}$$

$$= 1.03$$

b. Dynamic programming explanations—Refer to Page No. 265–266.

c. Formula for Floyd's approach to find shortest path—Refer to Page No. 275.

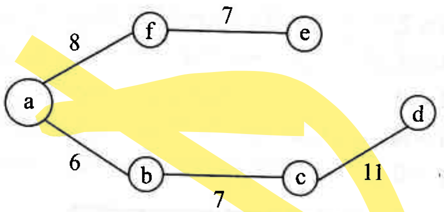
Solution for the given graph—Refer to Page No. 276, Figure 8.7.

7.

a. Kruskal's algorithm to find the cost for the given graph.

List the edges in a sorted order.

Edges	ab	bc	ef	af	ce	be	cd	de	cf	bf
Weight	6	7	7	8	9	10	11	13	14	20
Invertion status	✔	✔	✔	✔	x	x	✔	x	x	x
Inversion order	1	2	3	4	✔	✔	5	✔	✔	✔



Total cost = 39

b. Huffman tree construction algorithm—Refer to Page No. 311–312.

c. Decision tree for 3-elements invertion sort—Refer to Page No. 369, Figure 11.3.

8.

a. Difference between back tracking and branch-and-bound algorithm.

Back tracking	*Branch and bound algorithm*
(1) Non-optimization problems (2) Employs depth first search traversal (3) Requires a stack or use recursion	(1) Optimization problems (2) Employs best first search (not breath first search) traversal (3) We can use a heap to find the best node (the node with the best solution seen so far)

b. State space tree to generate first solution to 4-queen problem—Refer to Page No. 396, Figure 12.2.

Observe that, the solution is

	Q		
			Q
Q			
		Q	

Another solution is its minor image

		Q	
Q			
			Q
	Q		

c. P and NP problem—Refer to Page No. 373–374.

USN

06CS63

Fourth S

2009-

Time: 3 hrs

Note: 1. Answ

1.

a. Explain
chart.

b. Define th

i. Speci

ii. Paths

iii. Sets a

c. Write an
numbers.

2.

a. Prove that
If $t_1(n) \in g_1$
then $t_1($

b. Write an a
algorithm's

c. Explain the
limits. Com

3.

a. Discuss how
Draw the tre

b. What is stab

c. Write the al