1. What is recurrence for worst case of QuickSort and what is the time complexity in Worst case?

A Recurrence is T(n) = T(n-2) + O(n) and time complexity is O(n^2)

B Recurrence is T(n) = T(n-1) + O(n) and time complexity is O(n^2)

C Recurrence is T(n) = 2T(n/2) + O(n) and time complexity is O(nLogn)

D Recurrence is T(n) = T(n/10) + T(9n/10) + O(n) and time complexity is O(nLogn)

2. Suppose we have a O(n) time algorithm that finds median of an unsorted array. Now consider a QuickSort implementation where we first find median using the above algorithm, then use median as pivot. What will be the worst case time complexity of this modified QuickSort.

A O(n^2 Logn)

B O(n^2)

C O(n Logn Logn)

D O(nLogn)

3. Given an unsorted array. The array has this property that every element in array is at most k distance from its position in sorted array where k is a positive integer smaller than size of array. Which sorting algorithm can be easily modified for sorting this array and what is the obtainable time complexity?

A Insertion Sort with time complexity O(kn)

B Heap Sort with time complexity O(nLogk)

C Quick Sort with time complexity O(kLogk)

D Merge Sort with time complexity O(kLogk)

4. Which of the following is not true about comparison based sorting algorithms?

A The minimum possible time complexity of a comparison based sorting algorithm is O(nLogn) for a random input array

B Any comparison based sorting algorithm can be made stable by using position as a criteria when two elements are compared

C Counting Sort is not a comparison based sorting algortihm

D Heap Sort is not a comparison based sorting algorithm.

5 What is time complexity of fun()?

int fun(int n)

{

int count = 0;

for (int i = n; i > 0; i /= 2)

for (int j = 0; j < i; j++)

count += 1;

return count;

}

A O(n^2)

B O(nLogn)

C O(n)

D O(nLognLogn)

6 What is the time complexity of fun()?

int fun(int n)

{

int count = 0;

for (int i = 0; i < n; i++)

for (int j = i; j > 0; j--)

count = count + 1;

return count;

}

Run on IDE

A Theta (n)

B Theta (n^2)

C Theta (n\*Logn)

D Theta (nLognLogn)

7 The recurrence relation capturing the optimal time of the Tower of Hanoi problem with n discs is. A T(n) = 2T(n – 2) + 2

B T(n) = 2T(n – 1) + n

C T(n) = 2T(n/2) + 1

D T(n) = 2T(n – 1) + 1

8 Let w(n) and A(n) denote respectively, the worst case and average case running time of an algorithm executed on an input of size n. which of the following is ALWAYS TRUE? (GATE CS 2012)

(A) A(n) = \Omega(W(n))

(B) A(n) = \Theta(W(n))

(C) A(n) = O(W(n))

(D) A(n) = o(W(n))

A A

B B

C C

D D

9 Which of the following is not O(n^2)?

A (15^10) \* n + 12099

B n^1.98

C n^3 / (sqrt(n))

D (2^20) \* n

10 Which of the given options provides the increasing order of asymptotic complexity of functions f1, f2, f3 and f4?

f1(n) = 2^n

f2(n) = n^(3/2)

f3(n) = nLogn

f4(n) = n^(Logn)

A f3, f2, f4, f1

B f3, f2, f1, f4

C f2, f3, f1, f4

D f2, f3, f4, f1

11. Consider the following program fragment for reversing the digits in a given integer to obtain a new integer. Let n = D1D2…Dm

int n, rev;

rev = 0;

while (n > 0)

{

rev = rev\*10 + n%10;

n = n/10;

}

The loop invariant condition at the end of the ith iteration is: (GATE CS 2004)

A n = D1D2….Dm-i and rev = DmDm-1…Dm-i+1

B n = Dm-i+1…Dm-1Dm and rev = Dm-1….D2D1

C n != rev

D n = D1D2….Dm and rev = DmDm-1…D2D1

12 What is the best time complexity of bubble sort?

A N^2

B NlogN

C N

D N(logN)^2

13What is the worst case time complexity of insertion sort where position of the data to be inserted is calculated using binary search?

A N

B NlogN

C N^2

D N(logN)^2

14 The tightest lower bound on the number of comparisons, in the worst case, for comparison-based sorting is of the order of

A N

B N^2

C NlogN

D N(logN)^2

15 In a modified merge sort, the input array is splitted at a position one-third of the length(N) of the array. What is the worst case time complexity of this merge sort?

A N(logN base 3)

B N(logN base 2/3)

C N(logN base 1/3)

D N(logN base 3/2)

16 What is the time complexity of the below function?

void fun(int n, int arr[])

{

int i = 0, j = 0;

for(; i < n; ++i)

while(j < n && arr[i] < arr[j])

j++;

}

A O(n)

B O(n^2)

C O(nlogn)

D O(n(logn)^2)

17 In a competition, four different functions are observed. All the functions use a single for loop and within the for loop, same set of statements are executed. Consider the following for loops:

A) for(i = 0; i < n; i++)

B) for(i = 0; i < n; i += 2)

C) for(i = 1; i < n; i \*= 2)

D) for(i = n; i > -1; i /= 2) If n is the size of input(positive), which function is most efficient(if the task to be performed is not an issue)?

A A

B B

C C

D D

18 The following statement is valid. log(n!) = \theta(n log n).

A True

B False

19 What does it mean when we say that an algorithm X is asymptotically more efficient than Y?

A X will be a better choice for all inputs

B X will be a better choice for all inputs except small inputs

C X will be a better choice for all inputs except large inputs

D Y will be a better choice for small inputs

20 What is the time complexity of Floyd–Warshall algorithm to calculate all pair shortest path in a graph with n vertices?

A O(n^2logn)

B Theta(n^2logn)

C Theta(n^4)

D Theta(n^3)

21 A list of n string, each of length n, is sorted into lexicographic order using the merge-sort algorithm. The worst case running time of this computation is (A) O (n log n (B) O (n^2 log n) (C) O (n^2 + log n) (D) O (n^2)

A A

B B

C C

D D

22 In quick sort, for sorting n elements, the (n/4)th smallest element is selected as pivot using an O(n) time algorithm. What is the worst case time complexity of the quick sort? (A) \theta(n) (B) \theta(nLogn) (C) \theta(n^2) (D) \theta(n^2 log n)

A A

B B

C C

D D

23 Consider the Quicksort algorithm. Suppose there is a procedure for finding a pivot element which splits the list into two sub-lists each of which contains at least one-fifth of the elements. Let T(n) be the number of comparisons required to sort n elements. Then

A T(n) <= 2T(n/5) + n

B T(n) <= T(n/5) + T(4n/5) + n

C T(n) <= 2T(4n/5) + n

D T(n) <= 2T(n/2) + n

24 Consider the following functions:

f(n) = 2^n

g(n) = n!

h(n) = n^logn

Which of the following statements about the asymptotic behavior of f(n), g(n), and h(n) is true?

(A) f(n) = O(g(n)); g(n) = O(h(n))

(B) f(n) = \Omega(g(n)); g(n) = O(h(n))

(C) g(n) = O(f(n)); h(n) = O(f(n))

(D) h(n) = O(f(n)); g(n) = \Omega(f(n))

A A

B B

C C

D D

25 In the following C function, let n >= m.

int gcd(n,m)

{

if (n%m ==0) return m;

n = n%m;

return gcd(m, n);

}

How many recursive calls are made by this function?

(A) \theta(logn)

(B) \Omega(n)

(C) \theta(loglogn)

(D) \theta(sqrt(n))

A A

B B

C C

D D

26 Which of the following sorting algorithms has the lowest worst-case complexity?

A Merge Sort

B Bubble Sort

C Quick Sort

D Selection Sort

27

Consider the following functions formula Which of the following is true? (GATE CS 2000)

(a) h(n) is 0(f(n))

(b) h(n) is 0(g(n))

(c) g(n) is not 0(f(n))

(d) f(n) is 0(g(n))

A a

B b

C c

D d

28 Consider the following three claims I (n + k)^m = \theta(n^m), where k and m are constants II 2^(n + 1) = 0(2^n) III 2^(2n + 1) = 0(2^n) Which of these claims are correct? (GATE CS 2003)

A I and II

B I and III

C II and III

D I, II and III

29 Let s be a sorted array of n integers. Let t(n) denote the time taken for the most efficient algorithm to determined if there are two elements with sum less than 1000 in s. which of the following statements is true? (GATE CS 2000)

a) t (n) is 0 (1)

b) n < t (n) < n {log\_2 n}

c) n log 2 n < t (n) < {n \choose 2}

d) t (n) = {n \choose 2}

A a

B b

C c

D d

30 Consider the following function

int unknown(int n) {

int i, j, k = 0;

for (i = n/2; i <= n; i++)

for (j = 2; j <= n; j = j \* 2)

k = k + n/2;

return k;

}

What is the returned value of the above function? (A) \Theta(n^2)

(B) \Theta(n^2Logn)

(C) \Theta(n^3)

(D) \Theta(n^3Logn)

A A

B B

C C

D D

31 The number of elements that can be sorted in \Theta(logn) time using heap sort is

(A) \Theta(1)

(B) \Theta(\sqrt{logn})

(C) \Theta(Log n/(Log Log n))

(d) \Theta(Log n)

A A

B B

C C

D D

32 Consider the following two functions. What are time complexities of the functions?

int fun1(int n)

{

if (n <= 1) return n;

return 2\*fun1(n-1);

}

int fun2(int n)

{

if (n <= 1) return n;

return fun2(n-1) + fun2(n-1);

}

A O(2^n) for both fun1() and fun2()

B O(n) for fun1() and O(2^n) for fun2()

C O(2^n) for fun1() and O(n) for fun2()

D O(n) for both fun1() and fun2()

33 In quick sort, for sorting n elements, the (n/4)th smallest element is selected as pivot using an O(n) time algorithm. What is the worst case time complexity of the quick sort? <pre> (A) \theta(n) (B) \theta(nLogn) (C) \theta(n^2) (D) \theta(n^2 log n) </pre>

A A

B B

C C

D D

34 Consider the Quicksort algorithm. Suppose there is a procedure for finding a pivot element which splits the list into two sub-lists each of which contains at least one-fifth of the elements. Let T(n) be the number of comparisons required to sort n elements. Then

A T(n) <= 2T(n/5) + n

B T(n) <= T(n/5) + T(4n/5) + n

C T(n) <= 2T(4n/5) + n

D T(n) <= 2T(n/2) + n

35 Consider the following segment of C-code:

int j, n;

j = 1;

while (j <= n)

j = j\*2;

The number of comparisons made in the execution of the loop for any n > 0 is: Base of Log is 2 in all options.

A CEIL(logn) + 2

B n

C CEIL(logn)

D FLOOR(logn) + 2

36 Consider the following C-program fragment in which i, j and n are integer variables.

for (i = n, j = 0; i >0; i /= 2, j += i);

Let val(j) denote the value stored in the variable j after termination of the for loop. Which one of the following is true? (A) val(j) = \theta(logn) (B) vaI(j) = \theta(sqrt(n)) (C) val(j) = \theta(n) (D) val(j) = \theta(nlogn)

A A

B B

C C

D D

37 The minimum number of comparisons required to find the minimum and the maximum of 100 numbers is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

A 148

B 147

C 146

D 140

38 Consider the following pseudo code. What is the total number of multiplications to be performed?

D = 2

for i = 1 to n do

for j = i to n do

for k = j + 1 to n do

D = D \* 3

A Half of the product of the 3 consecutive integers.

B One-third of the product of the 3 consecutive integers.

C One-sixth of the product of the 3 consecutive integers.

D None of the above.

39 You have an array of n elements. Suppose you implement quicksort by always choosing the central element of the array as the pivot. Then the tightest upper bound for the worst case performance is

A O(n2)

B O(nLogn)

C Theta(nLogn)

D O(n3)

40 Consider the following C-function:

double foo (int n)

{

int i;

double sum;

if (n = = 0) return 1.0;

else

{

sum = 0.0;

for (i = 0; i < n; i++)

sum += foo (i);

return sum;

}

}

The space complexity of the above function is:

A O(1)

B O(n)

C O(n!)

D O(nn)

**UNIT - 3 (GREEDY )**

1. Which of the following standard algorithms is not a Greedy algorithm?

A Dijkstra's shortest path algorithm

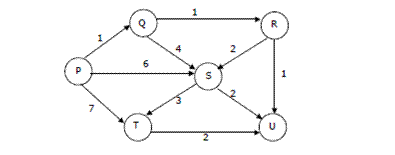
B Prim's algorithm

C Kruskal algorithm

D Huffman Coding

E Bellmen Ford Shortest path algorithm

2 Suppose we run Dijkstra’s single source shortest-path algorithm on the following edge weighted directed graph with vertex P as the source. In what order do the nodes get included into the set of vertices for which the shortest path distances are finalized?



A P, Q, R, S, T, U

B P, Q, R, U, S, T

C P, Q, R, U, T, S

D P, Q, T, R, U, S

3 A networking company uses a compression technique to encode the message before transmitting over the network. Suppose the message contains the following characters with their frequency:

character Frequency

a 5

b 9

c 12

d 13

e 16

f 45

Note : Each character in input message takes 1 byte. If the compression technique used is Huffman Coding, how many bits will be saved in the message?

A 224

B 800

C 576

D 324

4 What is the time complexity of Huffman Coding?

A O(N)

B O(NlogN)

C O(N(logN)^2)

D O(N^2)

5 In question #2, which of the following represents the word "dead"?

A 1011111100101

B 0100000011010

C Both A and B

D None of these

6 Which of the following is true about Kruskal and Prim MST algorithms? Assume that Prim is implemented for adjacency list representation using Binary Heap and Kruskal is implemented using union by rank.

A Worst case time complexity of both algorithms is same.

B Worst case time complexity of Kruskal is better than Prim

C Worst case time complexity of Prim is better than Kruskal

7 Which of the following is true about Huffman Coding.

A Huffman coding may become lossy in some cases

B Huffman Codes may not be optimal lossless codes in some cases

C In Huffman coding, no code is prefix of any other code.

D All of the above

8 Suppose the letters a, b, c, d, e, f have probabilities 1/2, 1/4, 1/8, 1/16, 1/32, 1/32 respectively. Which of the following is the Huffman code for the letter a, b, c, d, e, f?

A 0, 10, 110, 1110, 11110, 11111

B 11, 10, 011, 010, 001, 000

C 11, 10, 01, 001, 0001, 0000

D 110, 100, 010, 000, 001, 111

9 Suppose the letters a, b, c, d, e, f have probabilities 1/2, 1/4, 1/8, 1/16, 1/32, 1/32 respectively. What is the average length of Huffman codes?

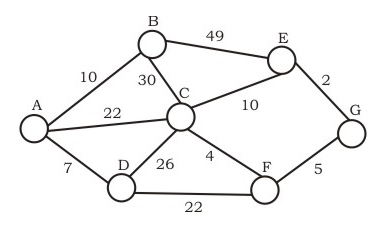
A 3

B 2.1875

C 2.25

D 1.9375

10 Consider the undirected graph below: primsMST Using Prim's algorithm to construct a minimum spanning tree starting with node A, which one of the following sequences of edges represents a possible order in which the edges would be added to construct the minimum spanning tree?



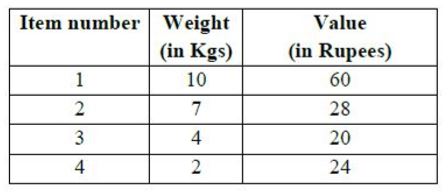
A (E, G), (C, F), (F, G), (A, D), (A, B), (A, C)

B (A, D), (A, B), (A, C), (C, F), (G, E), (F, G)

C (A, B), (A, D), (D, F), (F, G), (G, E), (F, C)

D (A, D), (A, B), (D, F), (F, C), (F, G), (G, E)

Consider the weights and values of items listed below. Note that there is only one unit of each item.



The task is to pick a subset of these items such that their total weight is no more than 11 Kgs and their total value is maximized. Moreover, no item may be split. The total value of items picked by an optimal algorithm is denoted by Vopt. A greedy algorithm sorts the items by their value-to-weight ratios in descending order and packs them greedily, starting from the first item in the ordered list. The total value of items picked by the greedy algorithm is denoted by Vgreedy. The value of Vopt − Vgreedy is \_\_\_\_\_\_ . Note -This was Numerical Type question.

A 16

B 8

C 44

D 60

12 A text is made up of the characters a, b, c, d, e each occurring with the probability 0.11, 0.40, 0.16, 0.09 and 0.24 respectively. The optimal Huffman coding technique will have the average length of:

A 2.40

B 2.16

C 2.26

D 2.15

13 Six files F1, F2, F3, F4, F5 and F6 have 100, 200, 50, 80, 120, 150 records respectively. In what order should they be stored so as to optimize act. Assume each file is accessed with the same frequency

A F3, F4, F1, F5, F6, F2

B F2, F6, F5, F1, F4, F3

C F1, F2, F3, F4, F5, F6

D Ordering is immaterial as all files are accessed with the same frequency

14 Which of the following standard algorithms is not Dynamic Programming based.

A Bellman–Ford Algorithm for single source shortest path

B Floyd Warshall Algorithm for all pairs shortest paths

C 0-1 Knapsack problem

D Prim's Minimum Spanning Tree

15 We use dynamic programming approach when

A It provides optimal solution

B The solution has optimal substructure

C The given problem can be reduced to the 3-SAT problem

D It's faster than Greedy

16 An algorithm to find the length of the longest monotonically increasing sequence of numbers in an array A[0 :n-1] is given below. Let Li denote the length of the longest monotonically increasing sequence starting at index i in the array. Which of the following statements is TRUE?

A The algorithm uses dynamic programming paradigm

B The algorithm has a linear complexity and uses branch and bound paradigm

C The algorithm has a non-linear polynomial complexity and uses branch and bound paradigm

D The algorithm uses divide and conquer paradigm.

17 Kadane algorithm is used to find:

A Maximum sum subsequence in an array

B Maximum sum subarray in an array

C Maximum product subsequence in an array

D Maximum product subarray in an array

18 Four matrices M1, M2, M3 and M4 of dimensions pxq, qxr, rxs and sxt respectively can be multiplied is several ways with different number of total scalar multiplications. For example, when multiplied as ((M1 X M2) X (M3 X M4)), the total number of multiplications is pqr + rst + prt. When multiplied as (((M1 X M2) X M3) X M4), the total number of scalar multiplications is pqr + prs + pst. If p = 10, q = 100, r = 20, s = 5 and t = 80, then the number of scalar multiplications needed is

A 248000

B 44000

C 19000

D 25000

19 The subset-sum problem is defined as follows. Given a set of n positive integers, S = {a1 ,a2 ,a3 ,…,an} and positive integer W, is there a subset of S whose elements sum to W? A dynamic program for solving this problem uses a 2-dimensional Boolean array X, with n rows and W+1 columns. X[i, j],1 <= i <= n, 0 <= j <= W, is TRUE if and only if there is a subset of {a1 ,a2 ,...,ai} whose elements sum to j. Which of the following is valid for 2 <= i <= n and ai <= j <= W?

A X[i, j] = X[i - 1, j] ∨ X[i, j -ai]

B X[i, j] = X[i - 1, j] ∨ X[i - 1, j - ai]

C X[i, j] = X[i - 1, j] ∧ X[i, j - ai]

D X[i, j] = X[i - 1, j] ∧ X[i -1, j - ai]

20 In the above question, which entry of the array X, if TRUE, implies that there is a subset whose elements sum to W?

A X[1, W]

B X[n ,0]

C X[n, W]

D X[n -1, n]

21 A sub-sequence of a given sequence is just the given sequence with some elements (possibly none or all) left out. We are given two sequences X[m] and Y[n] of lengths m and n respectively, with indexes of X and Y starting from 0. We wish to find the length of the longest common sub-sequence(LCS) of X[m] and Y[n] as l(m,n), where an incomplete recursive definition for the function l(i,j) to compute the length of The LCS of X[m] and Y[n] is given below:

l(i,j) = 0, if either i=0 or j=0

= expr1, if i,j > 0 and X[i-1] = Y[j-1]

= expr2, if i,j > 0 and X[i-1] != Y[j-1]

A expr1 ≡ l(i-1, j) + 1

B expr1 ≡ l(i, j-1)

C expr2 ≡ max(l(i-1, j), l(i, j-1))

D expr2 ≡ max(l(i-1,j-1),l(i,j))

22 Consider two strings A = "qpqrr" and B = "pqprqrp". Let x be the length of the longest common subsequence (not necessarily contiguous) between A and B and let y be the number of such longest common subsequences between A and B. Then x + 10y = \_\_\_.

A 33

B 23

C 43

D 34

23 Let A1, A2, A3, and A4 be four matrices of dimensions 10 x 5, 5 x 20, 20 x 10, and 10 x 5, respectively. The minimum number of scalar multiplications required to find the product A1A2A3A4 using the basic matrix multiplication method is

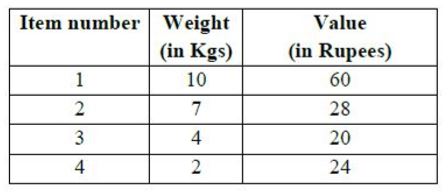
A 1500

B 2000

C 500

D 100

24 Consider the weights and values of items listed below. Note that there is only one unit of each

item. 

The task is to pick a subset of these items such that their total weight is no more than 11 Kgs and their total value is maximized. Moreover, no item may be split. The total value of items picked by an optimal algorithm is denoted by Vopt. A greedy algorithm sorts the items by their value-to-weight ratios in descending order and packs them greedily, starting from the first item in the ordered list. The total value of items picked by the greedy algorithm is denoted by Vgreedy. The value of Vopt − Vgreedy is \_\_\_\_\_\_ . Note -This was Numerical Type question.

A 16

B 8

C 44

D 60

25 Consider the following two sequences :

X = < B, C, D, C, A, B, C >, and

Y = < C, A, D, B, C, B >

The length of longest common subsequence of X and Y is :

A 5

B 3

C 4

D 2

26 Consider a sequence F00 defined as : F00(0) = 1, F00(1) = 1 F00(n) = 10 ∗ F00(n – 1) + 100 F00(n – 2) for n ≥ 2 Then what shall be the set of values of the sequence F00 ?

A (1, 110, 1200)

B (1, 110, 600, 1200)

C (1, 2, 55, 110, 600, 1200)

D (1, 55, 110, 600, 1200)

**unit - 2**

1 Which of the following algorithms is NOT a divide & conquer algorithm by nature?

A Euclidean algorithm to compute the greatest common divisor

B Heap Sort

C Cooley-Tukey fast Fourier transform

D Quick Sort

2 Consider the following C program

int main()

{

int x, y, m, n;

scanf ("%d %d", &x, &y);

/\* x > 0 and y > 0 \*/

m = x; n = y;

while (m != n)

{

if(m>n)

m = m - n;

else

n = n - m;

}

printf("%d", n);

}

What does the program compute?

A x + y using repeated subtraction

B x mod y using repeated subtraction

C the greatest common divisor of x and y

D the least common multiple of x and y

3 Consider the polynomial p(x) = a0 + a1x + a2x^2 +a3x^3, where ai != 0, for all i. The minimum number of multiplications needed to evaluate p on an input x is:

A 3

B 4

C 6

D 9

4Maximum Subarray Sum problem is to find the subarray with maximum sum. For example, given an array {12, -13, -5, 25, -20, 30, 10}, the maximum subarray sum is 45. The naive solution for this problem is to calculate sum of all subarrays starting with every element and return the maximum of all. We can solve this using Divide and Conquer, what will be the worst case time complexity using Divide and Conquer.

A O(n)

B O(nLogn)

C O(Logn)

D O(n^2)

5 Consider a situation where you don't have function to calculate power (pow() function in C) and you need to calculate x^n where x can be any number and n is a positive integer. What can be the best possible time complexity of your power function?

A O(n)

B O(nLogn)

C O(LogLogn)

D O(Logn)

6 Consider the problem of searching an element x in an array 'arr[]' of size n. The problem can be solved in O(Logn) time if. 1) Array is sorted 2) Array is sorted and rotated by k. k is given to you and k <= n 3) Array is sorted and rotated by k. k is NOT given to you and k <= n 4) Array is not sorted

A 1 Only

B 1 & 2 only

C 1, 2 and 3 only

D 1, 2, 3 and 4

7 The secant method is used to find the root of an equation f(x) = 0. It is started from two distinct estimates xa and xb for the root. It is an iterative procedure involving linear interpolation to a root. The iteration stops if f(xb) is very small and then xb is the solution. The procedure is given below. Observe that there is an expression which is missing and is marked by? Which is the suitable expression that is to be put in place of? So that it follows all steps of the secant method? Secant

Initialize: xa, xb, ε, N // ε = convergence indicator

fb = f(xb) i = 0

while (i < N and |fb| > ε) do

i = i + 1 // update counter

xt = ? // missing expression for

// intermediate value

xa = xb // reset xa

xb = xt // reset xb

fb = f(xb) // function value at new xb

end while

if |fb| > ε

then // loop is terminated with i = N

write “Non-convergence”

else

write “return xb”

end if

A xb – (fb– f(xa)) fb/ (xb – xa)

B xa– (fa– f(xa)) fa/ (xb – xa)

C xb – (fb – xa) fb/ (xb – fb(xa)

D xa – (xb – xa) fa/ (fb – f(xa))

8 Suppose you are provided with the following function declaration in the C programming language.

int partition (int a[], int n);

The function treats the first element of a[] as a pivot, and rearranges the array so that all elements less than or equal to the pivot is in the left part of the array, and all elements greater than the pivot is in the right part. In addition, it moves the pivot so that the pivot is the last element of the left part. The return value is the number of elements in the left part. The following partially given function in the C programming language is used to find the kth smallest element in an array a[ ] of size n using the partition function. We assume k ≤ n

int kth\_smallest (int a[], int n, int k)

{

int left\_end = partition (a, n);

if (left\_end+1==k)

{

return a [left\_end];

}

if (left\_end+1 > k)

{

return kth\_smallest (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_);

}

else

{

return kth\_smallest (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_);

}

}

The missing argument lists are respectively

A (a, left\_end, k) and (a+left\_end+1, n–left\_end–1, k–left\_end–1)

B (a, left\_end, k) and (a, n–left\_end–1, k–left\_end–1)

C (a, left\_end+1, N–left\_end–1, K–left\_end–1) and(a, left\_end, k)

D (a, n–left\_end–1, k–left\_end–1) and (a, left\_end, k)

9 Consider the problem of computing min-max in an unsorted array where min and max are minimum and maximum elements of array. Algorithm A1 can compute min-max in a1 comparisons without divide and conquer. Algorithm A2 can compute min-max in a2 comparisons by scanning the array linearly. What could be the relation between a1 and a2 considering the worst case scenarios?

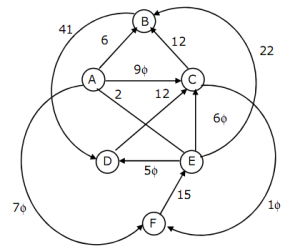
A a1 < a2

B a1 > a2

C a1 = a2

D Depends on the input

10 Let G be the directed, weighted graph shown in below figure



We are interested in the shortest paths from A.

a) Output the sequence of vertices identified by the Dijkstra’s algorithm for single source shortest path when the algorithm is started at node A.

b) Write down sequence of vertices in the shortest path from A to E.

c) What is the cost of the shortest path from A to E?

**unit - 3,4 & 5**

131) there are 5 items as follows

|  |  |  |
| --- | --- | --- |
| Items | wi | vi |
| Item1 | 5 pounds | 30$ |
| Item2 | 10 pounds | 20$ |
| Item3 | 20 pounds | 100$ |
| Item4 | 30 pounds | 90$ |
| Item5 | 40 pounds | 160$ |

The knapsack can hold 60 pounds find the optimal solution

(A)   250$                (B) 260 $         (C) 270 $                      (D) 290$

(132) there are 5 items as follows

|  |  |  |
| --- | --- | --- |
| Items | wi | vi |
| Item1 | 5 pounds | 30$ |
| Item2 | 10 pounds | 20$ |
| Item3 | 20 pounds | 100$ |
| Item4 | 30 pounds | 90$ |
| Item5 | 40 pounds | 160$ |

The knapsack can hold 60 pounds find the solution by greedy technique

(A)   230$                (B) 260 $         (C) 220 $                      (D) 250$

(133) what is an optimal Huffman code for alphabet**a** of the following set of frequencies a: 05, b:48, c:07, d:17, e:10, f:13

(A) 1010                      (B)0101                       (C) 1001                      (D) 1100

(134) the total running time of Huffman on the set of n characters is

(A) O(n)                       (B) O(n log n)              (C) O(n2)                      (D) O(log n)

(135) the total running time of matrix chain multiplication of n matrices

(A) ϴ (n4)         (B) ϴ (n3)**(**C) ϴ (n2)         (D) ϴ (n)

(136) which of the following is true

(A) P is subset of NP                            (B) NP is subset of P

(C) P and NP are equal                       (D) NP is subset of NP hard

(137) the total running time of optimal binary search tree of n nodes

(A) O(n2)                      (B) O(n)                       (C) O(n3)                      (D) O(n log n)

(138) If every square of the board is visited, then the total number of knight moves of n-queen problem is

(A) n3-1            (B) n-1                         (C)n2-1                         (D) log n-1

(139) If every square of the board is visited, then the total number of knight moves of 4-queen problem is

(A) 14              (B) 15                          (C) 16                          (D) 12

(140) If every square of the board is visited, then the total number of knight moves of 8-queen problem is

(A) 64              (B) 62                          (C) 61                          (D) 63

(141) In which of the following cases n-queen problem does not exist

(A) n=2 and n=4          (B) n=4 and n=6          (C) n=2 and n=3          (D) n=4 and n=8

(142) the total running time of knapsack problem for a simple approach

(A) O(n)                       (B) O( log n)                (C) O(2n log n)                         (D) O(2n)

(143) what is an optimal Huffman code for alphabet**a** of the following set of frequencies a: 01, b:01, c:02, d:03, e:05, f:8, g:13, h:21

(A) 001010                  (B) 001111                  (C) 111100                  (D) 101010

(144)  what is an optimal Huffman code for alphabet**b** of the following set of frequencies a: 45, b:13, c:12, d:16, e:9, f:5

(A) 100                        (B) 111                        (C) 001                        (D) 101

(145) what is an optimal Huffman code for alphabet**e** of the following set of frequencies a: 29, b:25, c:20, d:12, e:05, f:09

(A) 100 0                      (B) 1110                                  (C) 0010                                  (D) 1011

(146) Which of the following method is taking overcharge for some operations in amortized analysis?

(A) Aggregate method                        (B) accounting method

(C) potential method              (D) both (A) and (C)

(147) Which of the following method is most flexible in amortized analysis?

(A) Aggregate method                        (B) accounting method

(C) potential method              (D) both (A) and (B)

(148) Which of the following method is taken different operations different charges in amortized analysis?

 (A) Aggregate method                       (B) accounting method

(C) potential method                          (D) both (A) and (B)

(149) Which of the following method is computing total cost of an algorithm in amortized analysis?

(A) Aggregate method                        (B) accounting method

(C) potential method              (D) both (C) and (B)

(150) which of the following method is credit as the potential energy to pay for future operations?

(A) Aggregate method                        (B) accounting method

(C) potential method              (D) both (A) and (B)

(151) If all c(i, j )’s and r(i, j)’s are calculated, then OBST algorithm in worst case takes one of the following time.

(a)  O(n log n)           
(b)  O(n3)                  
(c)  O(n2)                   
(d)  O(log n)              
(e)  O(n4).

(152) The following is a weighted binary tree, then what is the weighted array for the TVS problem?

(a)  [9, 2, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 6, 4]    
(b)  [9, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 7, 4, 6]

(c)  [9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 6, 7, 4]    
(d)  [9, 2, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 6, 4]

(e)  [9, 2, 0, 0, 0, 7, 0, 0, 0, 0, 6, 4, 0, 0]

(153) The upper bound on the time complexity of the nondeterministic sorting algorithm is

(a)  O(n)                   
(b)  O(n log n)           
(c)  O(1)                   
(d)  O( log n)          

(154) The worst case time complexity of the nondeterministic dynamic knapsack algorithm is

(a)  O(n log n)                                         
(b)  O( log n)            
(c)  O(n2)                  
(d)  O(n)                 

(155) The time complexity of the normal quick sort, randomized quick sort algorithms in the worst case is

(a)  O(n2), O(n log n)                  (b)  O(n2), O(n2)

(c)  O(n log n), O(n2)                  (d)  O(n log n), O(n log n)

(156) Let there be an array of length ‘N’, and the selection sort algorithm is used to sort it, how many times a swap function is called to complete the execution?

(a)  N log N times                                  (b)  log N times

(c)  N2 times                                       (d)  N-1 times

(157) The Sorting method which is used for external sort is

(a)  Bubble sort                 (b)  Quick sort                       (c)  Merge sort        (d)  Radix sort

(158) In analysis of algorithm, approximate relationship between the size of the job and the amount of work required to do is expressed by using \_\_\_\_\_\_\_\_\_

(a)   Central tendency

(b) Differential equation

(c)   Order of execution                          (d) Order of magnitude

(159) P, Q and R are pointer variables. The statements below are intended to swap the contents of the nodes pointed to by P and Q. rewrite it so that it will work as intended.  
P = Q;    R = Q;   Q = R;

(a)   R=Q;  P=R;   Q=R;                          (b)   R=P;   P=P;   Q=Q;

(c)   P=P;   P=Q;   R=Q;                         (d)   R=P;   P=Q;   Q=R;

(160) Consider the usual algorithm for determining whether a sequence of parentheses is balanced. What is the maximum number of parentheses that will appear on the stack AT ANY ONE TIME when the algorithm analyzes: (()(())(()))

(a) 1                        (b) 2                        (c) 3                        (d) 4

 (161) The Knapsack problem where the objective function is to minimize the profit is \_\_\_\_\_\_

(a)   Greedy                                          (b)  Dynamic 0 / 1

(c)  Back tracking                                  (d)   Branch & Bound 0/1

(162) Choose the correct answer for the following statements:

I.     The theory of NP–completeness provides a method of obtaining a polynomial time for NPalgorithms.

II.     All NP-complete problem are NP-Hard.

(a)   I is FALSE and II is TRUE

(b)   I is TRUE and II is FALSE

(c)   Both are TRUE

(d)   Both are FALSE

(163) For 0/1 KNAPSACK problem, the algorithm takes \_\_\_\_\_\_\_\_ amount of time for memory table, and \_\_\_\_\_\_time to determine the optimal load, for N objects and W as the capacity of KNAPSACK.

(a)   O(N+W), O(NW)      (b)  O(NW), O(N+W)

(c)   O(N), O(NW)           (d)  O(NW), O(N)

(164) What is the type of the algorithm used in solving the 8 Queens problem?

(a)Greedy

(b)Dynamic

(c)Branch and Bound

(d)Backtracking.

(165) Sorting is not possible by using which of the following methods?

(a)Insertion

(b)Selection

(c)Deletion

(d)Exchange

**Algorithms Design And Analysis MCQS with Answers**

|  |  |
| --- | --- |
| 130 | B |
| 131 | B |
| 132 | C |
| 133 | A |
| 134 | B |
| 135 | B |
| 136 | A |
| 137 | C |
| 138 | C |
| 139 | B |
| 140 | D |
| 141 | C |
| 142 | D |
| 143 | C |
| 144 | D |
| 145 | B |
| 146 | B |
| 147 | C |
| 148 | B |
| 149 | A |
| 150 | C |

|  |  |
| --- | --- |
| 151 | B |
| 152 | D |
| 153 | A |
| 154 | D |
| 155 | B |
| 156 | D |
| 157 | C |
| 158 | D |
| 159 | D |
| 160 | C |
| 161 | D |
| 162 | A |
| 163 | B |
| 164 | D |
| 165 | C |

**Unit - 2**

111) If each node in a tree has value greater than every value in its left sub tree and value less than every value in its right sub tree, the tree is known as

(A) complete tree                   (B) full binary tree

(C) binary search tree                         (D) threaded tree

(112) Which of the following sorting procedure is the slowest?

(A) Quick sort              (B) Heap sort               (C) Shell sort               **(D) Bubble sort**

(113) which of the following shows the correct relationship among some of the more common computing times on algorithms

(A) O(log n) < O(n) < O( n\* log n) < O(2n) < O(n2)

(B) O(n) < O(log n) < O( n\* log n) < O(2n) < O(n2)

(C) O(n) < O(log n) < O( n\* log n) < O(n2) < O(2n)

(D) O(log n) < O(n) < O( n\* log n) < O(n2) < O(2n)

(114) The average time required to perform a successful sequential search for an element in an array A(1..n) is given by

(A) (n+1)/2                  (B) n(n+1)/2                (C) log n                      (D) n2

(115) the time complexity of linear search algorithm over an array of n elements is

(A) O(log n)                 (B) O(n)                       (C)  O( n log n)            (D) O(n2)

(116) the time taken by binary search algorithm to search a key in a sorted array of n elements is

(A) O(log n)                 (B) O(n)                       (C)  O( n log n)            (D) O(n2)

(117) the time required to search an element in a linked list of length n is

(A) O(log n)                 (B) O(n)                       (C)  O( 1)                     (D) O(n2)

(118) the worst case time required to search a given element in sorted linked list of length n is

(A) O(1)                       (B) O( log n)                (C)  O(n)          (D) O(n log n)

(119)  consider a linked list of n elements which is pointed by an external pointer. What is the time taken to delete the element which is successor of the element pointed to by a given pointer?

(A) O(1)                       (B) O( log n)                (C)  O(n)          (D) O(n log n)

(120) consider a linked list of n elements. What is the time taken to insert an element an after element pointed by some pointer?

(A) O(1)                       (B) O( log n)                (C)  O(n)          (D) O(n log n)

 (121) which of the following operations is performed more efficiently by doubly linked list than by linear linked list?

(A) Deleting a node whose location is given

(B) searching an unsorted list for a given item

(C) inserting a node after the node with a given location

(D) Traversing the list to process each node

(122) the five items: A,B,C,D and E are pushed in a stack, one after the other starting from A. The stack is popped four items and each element is inserted in a queue. Then two elements are deleted from the queue and pushed back on the stack. Now on item is popped from the stack. The popped item is

(A) A                (B) B                (C) C                (D) D

(123) the time required to search an element in a binary search tree having n elements is

(A) O(1)                       (B) O( log n)                (C)  O(n)          (D) O(n log n)

(124) for a linear search in an array of n elements the time complexity for best, worst and average case are …., and …respectively.

(A) O(n) , O(1) and O(n/2)                              (B)  O(1) , O(n) and O(n/2)

(C) ) O(1) , O(n) and O(n)                               (D) ) O(1) , O(n) and O((n-1)/2)

(125) the number of comparisons required by binary search of 100000 elements is

(A) 15              (B) 20              (C) 25              (D) 30

(126) Find an optimal parenthesization of a  matrix chain product whose sequence of dimension s is <5,4,6,2,7>

(A) 156            (B) 154            (C) 158            (D) 157

(127) Find an optimal parenthesization  of a  matrix chain product whose sequence of dimension s is <5,10,3,12, 5, 50, 6>

(A) 2010          (B) 2020          (C) 2015          (D) 2030

(128) Find an optimal parenthesization  of a  matrix chain product whose sequence of dimension s is <4,10,3,12,20,7>

(A) 1334          (B) 1324          (C)1344           (D)1354

(129) Find an optimal parenthesization  of a  matrix chain product whose sequence of dimension s is <5,4,3>  (for three matrices)

(A) 125            (B) 130            (C) 135            (D) 140

(130) Find an optimal parenthesization  of a  matrix chain product whose sequence of dimension s is <30,35,15,5,10,20,25> (for six matrices)

(A)7130           (B) 7125          (C) 7145          (D) 7135

**Algorithms Design And Analysis MCQS with Answers**

|  |  |
| --- | --- |
| 111 | C |
| 112 | D |
| 113 | D |
| 114 | A |
| 115 | B |
| 116 | A |
| 117 | B |
| 118 | C |
| 119 | A |
| 120 | A |
| 121 | A |
|  |  |
| 122 | D |
| 123 | B |
| 124 | C |
| 125 | B |

|  |  |
| --- | --- |
| 126 | C |
| 127 | A |
| 128 | C |
| 129 | C |
| 130 | B |

unit - 3

(76) Struct x

            {

            int i;

            char c;

            }

             union y{

            struct x a;

            double d;

            };

             printf("%d",sizeof(union y));

   (A)8                    (B)5                        (C)4                (D)1

(77) Worst case complexity of the insertion sort algorithm is   
(A) O(n2)          (B) O(n)           (C) O(n-1)        (D) O(n+1)  
(78) Average case complexity of the insertion sort algorithm is  
(A) O(n2)          (B) O(n)           (C) O(n-1)        (D) O(n+1)  
(79) Best case complexity of the insertion sort algorithm is  
(A) O(n2)          (B) O(n)           (C) O(n-1)        (D) O(n+1)  
  
(80)  Worst case complexity of the bubble sort algorithm is   
(A) O(n3)          (B) O(n4)          (C) O(n2)          (D) O(n)  
  
(81) Best case complexity of the bubble sort algorithm is   
(A) O(n3)          (B) O(n4)          (C) O(n2)          (D) O(n)  
(82) Average case complexity of the bubble sort algorithm is  
(A) O(n3)          (B) O(n4)          (C) O(n2)          (D) O(n)  
(83) Worst case complexity of the selection sort algorithm is  
(A) O(n3)          (B) O(n4)          (C) O(n2)          (D) O(n)  
(84) Average case complexity of the selection sort algorithm is  
(A) O(n3)          (B) O(n4)          (C) O(n2)          (D) O(n)  
(85) Best case complexity of the selection sort algorithm is  
(A) O(n3)          (B) O(n4)          (C) O(n2)          (D) O(n)  
(86)  If a complete binary tree Tn has n=1000 nodes then its height is  
(A) 21              (B) 10              (C) 11              (D) 12  
(87) If a complete binary tree Tnhas n=1000000 nodes then its height is  
(A) 21              (B) 20              (C) 23              (D) 22  
(88) The running time of Strassen’s algorithm for matrix multiplication is  
(A) ϴ (n)          (B) ϴ (n3)**(**C) ϴ (n2)         (D) ϴ (n2.81)  
(89) The running time of Floyd-Warshall algorithm is  
(A) ϴ (n)          (B) ϴ (n3)**(**C) ϴ (n2)         (D) ϴ (n log n)  
(90) Dijkastra’s algorithm bears some similarity to   
(A) BFS             (B) prim’s algorithm               (C) DFS            (D) Both (A) & (C)

### ALGORITHMS DESIGN AND ANALYSIS MCQS WITH ANSWERS

|  |  |
| --- | --- |
| 76 | A |
| 77 | A |
| 78 | A |
| 79 | C |
| 80 | C |
| 81 | C |
| 82 | C |
| 83 | C |
| 84 | C |
| 85 | C |
| 86 | C |
| 87 | A |
| 88 | D |
| 89 | C |
| 90 | D |

**unit - 2**

(46) If Total complexity after micro analysis is 5n3 + 10n2 + 100 n +400 logn+ 10,

The Big Oh complexity is

(A)   O(n2)    (B) O(n3)          (C) O(nlogn)    (D) O(n2logn)

(47) In Strassen’s Multiplication Algorithm the T(n)  is

A) 7T (n) + bn2              B) 7T (n/2) + bn2      C) 8T (n/2) + bn2               D) 7T (n/2) + bn

(48) T (n) = 4 T (n/2) + n   then in Big Oh Notation it is

A)  O (n2)                    B) O(4)                       (C)  O(n)                      D) O(log(n))

(49) In T(n) = a \* T(n/b) + f(n)  ,  a refers to

     (A)  Size of sub problem     (B) No. of sub problems

    (C) Size of the problem       (D) Time to combine solutions

(50) 0-1 knapsack be solved using

(A)   dynamic programming                       (B)  Backtracking        (C) Branch & Bound

(D) All A,B,C,E                               (E) Genetic Programming

(51) In depth first search algorithm the no. of recursive calls we have to make are

      (A) 2             (B) 1             (C)   6            (D) depends on the graph

(52) O (f(n))  minus O(f(n))  is equal to

(A)  Zero                 (B) A constant             (C) f(n)             (D) O(f(n))

(53) Quick sort is solved using

    (A) Divide and conquer                   (B) Greedy Programming

    (C) Dynamic Programming                         (D) Branch and bound

(54) For i = 1 to n-1 do

2.1 For j = 1 to n-1-i do

2.2.1 If (a[j+1] < a[j]) then swap a[j] and a[j+1]

Given code is for

(A)  Bubble sort        (B) Insertion sort               (C) Quick Sort                    (D) Selection Sort

(55) Worst case complexity of quick sort is

   (A)  O(n)                               (B) O(logn)                         (C) O(nlogn )                            (D) O(n2)

(56) The sub problems in Divide and Conquer are considered to be

A) Distinct         (B) overlapping                        (C) large size                      (D) small size

(57) Which of the following name does not relate to stacks?

(A)  FIFO lists               (B) LIFO list                 (C) Piles                                   (D) Push-down lists

(58) Which of the following data structure is linear type?

(A) Strings       (B) Lists                       (C) Queues      D) All of above

(59) In a graph if e=(u, v) means

(A)  u is adjacent to v but v is not adjacent to u                     (B) e begins at u and ends at v

(C)  u is processor and v is successor                         (D) both b and c

(60) An algorithm that calls itself directly or indirectly is known as

(A) Sub algorithm        (B) Recursion  (C) Polish notation      (D) Traversal algorithm

**Algorithms Design And Analysis MCQS with Answers**

|  |  |
| --- | --- |
| 46 | B |
| 47 | B |
| 48 | A |
| 49 | B |
| 50 | D |

|  |  |
| --- | --- |
| 51 | D |
| 52 | D |
| 53 | A |
| 54 | A |
| 55 | D |
| 56 | D |
| 57 | A |
| 58 | D |
| 59 | D |
| 60 | B |