ALGORITHMS

Solutions

1. Which of the following is not an in-place algorithm?

(a) Insertion sort
(b) Selection sort
(c) Merge sort
(d) Heap sort

Solution: Option (c)

Explanation:
An in-place algorithm is an algorithm which uses a constant amount of extra space apart from input. Merge sort uses an extra O (n) space in the merging part.

2. Which of the following is not a backtracking algorithm?

(a) Knight tour problem
(b) N queen problem
(c) Tower of Hanoi
(d) M coloring problem

Solution: Option (c)

3. Given two arrays of numbers a_1, a_2, a_3,…an and b_1, b_2, .. b_n where each number is 0 or 1, the fastest algorithm to find the largest span(i, j) such that a_i + a_{i+1}, ….a_j = b_i, b_{i+1}, .. b_j. or report that there is not such span,

(a) Takes O (n_3) and Ω (2n) time if hashing is permitted
(b) Takes O (n_3) and Ω (n2.5) time in the key comparison model
(c) Takes θ (n) time and space
(d) Takes O (√n) time only if the sum of the 2n elements is an even number

Solution: Option (c)

4. What is the return value of following function for arr[ ] = {9, 12, 2, 11, 2, 2, 10, 9, 12, 10, 9, 11, 2} and n is size of this array?

```
int fun(int arr[ ], int n)
{
   int x = arr[0];
   for (int i = 1; i < n; i++)
      x = x ^ arr[i];
```

```
    return x;
}
```

(a) 0
(b) 9
(c) 12
(d) 2

Solution: Option (b)

Explanation:
Note that 9 is the only element with odd occurrences, all other elements have even occurrences. If the input array has all elements with even occurrences except one, then the function returns the only element with odd occurrences. Note that XORing an element with itself results 0 and XOR of 0 with a number x is equal to x.

5. What does the following C expression do?

```
x = (x<<1) + x + (x>>1);
```

(a) Multiplies an integer with 7
(b) Multiplies an integer with 3.5
(c) Multiplies an integer with 3
(d) Multiplies an integer with 8

Solution: Option (b)

Explanation:
The expression multiplies an integer with 3.5. For example, if x is 4, the expression returns 15. If x is 6, it returns 21. If x is 5, it returns 17.

6. What does the following C expression do?

```
x = x & (x-1)
```

(a) Sets all bits as 1
(b) Makes x equals to 0
(c) Turns of the rightmost set bit
(d) Turns of the leftmost set bit

Solution: Option (c)

Explanation:
The expression simply turns off the rightmost set bit. For example, if x = 14 (1110), it returns 12 (1100).

7. Consider the C function given below. Assume that the array listA contains n (> 0) elements, sorted in ascending order.

```
int ProcessArray(int *listA, int x, int n)
{
   int i, j, k;
    i = 0;
    j = n-1;
    do
     {
        k = (i+j)/2;
if (x ≤ listA[k])
    j = k-1;
        if (listA[k] ≤ x)
            i = k+1;
    }
while (i ≤ j);
    if (listA[k] = = x)
return(k);
    else
return -1;
}
```

Which one of the following statements about the function ProcessArray is CORRECT?

(a) It will run into an infinite loop when x is not in listA.
(b) It is an implementation of binary search.
(c) It will always find the maximum element in listA.
(d) It will return −1 even when x is present in listA.

Solution: Option (b)

8. What is the output of following program?

```
#include <stdio.h>
void print(int n, int j)
{
   if (j ≥ n)
      return;
   if (n-j > 0 && n-j ≥ j)
       printf("%d %d\n", j, n-j);
      print(n, j+1);
}

int main()
{
   int n = 8;
   print(n, 1);
 }
```

(a) 1 7
    2 6
    3 5
    4 4
    44

(b) 1 7
    2 6
    3 5
    4 4

(c) 1 7
    2 6
    3 5

(d) 1 2
    34
    56
    78

Solution: Option (b)

Explanation:
For a given number n, the program prints all distinct pairs of positive integers with sum equal to n.

9. Which of the following is correct recurrence for worst case of Binary Search?

(a) $T(n) = 2T(n/2) + O(1)$ and $T(1) = T(0) = O(1)$
(b) $T(n) = T(n-1) + O(1)$ and $T(1) = T(0) = O(1)$
(c) $T(n) = T(n/2) + O(1)$ and $T(1) = T(0) = O(1)$
(d) $T(n) = T(n-2) + O(1)$ and $T(1) = T(0) = O(1)$

Solution: Option (c)

10. Given a sorted array of integers, what can be the minimum worst case time complexity to find ceiling of a number x in given array? Ceiling of an element x is the smallest element present in array which is greater than or equal to x. Ceiling is not present if x is greater than the maximum element present in array. For example, if the given array is {12, 67, 90, 100, 300, 399} and x = 95, then output should be 100.

(a) O (Log Log n)
(b) O (n)
(c) O (Log n)
(d) O (Log n * Log n)

Solution: Option (c)

Explanation:
We modify standard binary search to find ceiling. The time complexity T(n) can be written as:
$T(n) \leq T(n/2) + O(1)$

Solution of above recurrence can be obtained by Master Method. It falls in case 2 of Master Method. Solution is O (Log n).

11. Consider the following C program that attempts to locate an element x in an array Y[] using binary search. The program is erroneous. (GATE CS 2008)

```
f(int Y[10], int x) {
  int i, j, k;
  i = 0; j = 9;
  do {
          k =  (i + j) /2;
          if( Y[k] < x)  i = k; else j = k;
      } while(Y[k] != x && i < j);
  if(Y[k] = = x) printf ("x is in the array ") ;
  else printf (" x is not in the array ") ;
}
```

On which of the following contents of Y and x does the program fail?

(a) Y is [1 2 3 4 5 6 7 8 9 10] and x < 10
(b) Y is [1 3 5 7 9 11 13 15 17 19] and x < 1
(c) Y is [2 2 2 2 2 2 2 2 2 2] and x > 2
(d) Y is [2 4 6 8 10 12 14 16 18 20] and 2 < x < 20 and x is even

Solution: Option (c)

Explanation:
The above program doesn't work for the cases where element to be searched is the last element of Y[ ] or greater than the last element (or maximum element) in Y[ ]. For such cases, program goes in an infinite loop because i is assigned value as k in all iterations, and i never becomes equal to or greater than j. So while condition never becomes false.

12. In the above question, the correction needed in the program to make it work properly is: (GATE CS 2008)

(a) Change line 6 to: if (Y[k] < x) i = k + 1; else j = k-1;
(b) Change line 6 to: if (Y[k] < x) i = k - 1; else j = k+1;
(c) Change line 6 to: if (Y[k] ≤ x) i = k; else j = k;
(d) Change line 7 to: } while ((Y[k] = = x) && (i < j));

Solution: Option (a)

Explanation:
Below is the corrected function:

```
f(int Y[10], int x) {
int i, j, k;
i = 0; j = 9;
do {
k =  (i + j) /2;
if( Y[k] < x)  i = k + 1; else j = k - 1;
      } while(Y[k] != x && i < j);
if(Y[k] = = x) printf ("x is in the array ") ;
   else printf (" x is not in the array ") ;
}
```

13. You are given a list of 5 integers and these integers are in the range from 1 to 6. There are no duplicates in list. One of the integers is missing in the list. Which of the following expression would give the missing number?

^ is bitwise XOR operator.
~ is bitwise NOT operator.

Let elements of list can be accessed as list[0], list[1], list[2], list[3], list[4]

(a) list[0] ^ list[1] ^ list[2] ^ list[3] ^ list[4]
(b) list[0] ^ list[1] ^ list[2] ^ list[3] ^ list[4] ^ 1 ^ 2 ^ 3 ^ 4 ^ 5 ^ 6
(c) list[0] ^ list[1] ^ list[2] ^ list[3] ^ list[4] ^ 1 ^ 2 ^ 3 ^ 4 ^ 5
(d) ~(list[0] ^ list[1] ^ list[2] ^ list[3] ^ list[4])

Solution: Option (b)

Explanation:
XOR of all list elements and numbers from 1 to 6 gives the missing number.

14. In a village, people build houses in the same side of the road. A thief plans to loot the village. He wants maximum amount of money without having any risk of getting caught. By some means, the villagers know that their adjacent house is being looted or not and thus they become alert. So the thief cannot loot contiguous two houses. Given that the thief knows the amount of money stored in each house and the road is straight and there is no turning, which is the most efficient algorithmic strategy to solve this problem?

(a) Brute-force
(b) Dynamic Programming
(c) Backtracking
(d) Divide and Conquer

Solution: Option (b)

15. 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]

Solution: Option (c)

Explanation:
If we get the entry X[n, W] as true then there is a subset of {a_1, a_2, .. a_n} that has sum as W.

16. 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))

Solution: Option (c)

17. 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

Solution: Option (d)

Explanation:
The LCS is of length 4. There are 3 LCS of length 4 “qprr”, “pqrr” and “qpqr”.

18. 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

Solution: Option (b)

19. Four matrices M_1, M_2, M_3 and M_4 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 $((M_1 \times M_2) \times (M_3 \times M_4))$, the total number of multiplications is pqr + rst + prt. When multiplied as $(((M_1\times M_2) \times M_3) \times M_4)$, 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

Solution: Option (c)

Explanation:
It is basically matrix chain multiplication problem. We get minimum number of multiplications using $((M_1 \times (M_2 \times M_3)) \times M_4)$.

Total number of multiplications = 100×20×5 (for M2 × M3) + 10×100×5 + 10×5×80 = 19000.

20. The subset-sum problem is defined as follows. Given a set of n positive integers, S= {a_1, a_2, a_3,…, a_n} 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],$1 \leq i \leq n$, $0 \leq j \leq W$, is TRUE if and only if there is a subset of {a_1 ,a_2 ,...,a_i} whose elements sum to j.

Which of the following is valid for $2 \leq i \leq n$ and $a_i \leq j \leq W$?

(a) $X[i, j] = X[i – 1, j] \vee X[i, j -a_i]$
(b) $X[i, j] = X[i – 1, j] \vee X[i – 1, j – a_i]$
(c) $X[i, j] = X[i – 1, j] \vee X[i, j – a_i]$
(d) $X[i, j] = X[i – 1, j] \vee X[i -1, j – a_i]$

Solution: Option (b)

Explanation:
X[i, j] ($2 \leq i \leq n$ and $a_i \leq j \leq W$), is true if any of the following is true

1) Sum of weights excluding a_i is equal to j, i.e., if X[i-1, j] is true.
2) Sum of weights including a_i is equal to j, i.e., if X[i-1, j-a_i] is true so that we get $(j – a_i) + a_i$ as j.