Q.1) Max Marks: 1

Which of the following identities are false

A $(r_1^*)^* \equiv r_1^*$

B $r_1^*(r_1 + r_2)^* \equiv (r_1 + r_2)^*$

C $(r_1 + r_2)^* \equiv (r_1 r_2)^*$ Correct Option

Solution: (C)

Explanation:

I. $(r_1^*)^* \equiv r_1^* \Rightarrow$ True

Let $r_1 = a \Rightarrow r_1^* = a^* \Rightarrow (r_1^*)^* = (a^*)^* = a^* = r_1^*$

II. $r_1^*(r_1 + r_2)^* \equiv (r_1 + r_2)^* \Rightarrow$ True

Let $r_1 = a$, and $r_2 = b \Rightarrow r_1^*(r_1 + r_2)^* = a^*(a+b)^* = \{\varepsilon, a, aa, aaa, ...\}.\{\varepsilon, a, b, aa, ab, ba, bb, ...\}$

$= \{\varepsilon, a, b, aa, ab, ba, bb, ...\}$

III. $(r_1 + r_2)^* \equiv (r_1 r_2)^* \Rightarrow$ False

Let $r_1 = a$, and $r_2 = b$

$(r_1 + r_2)^* = (a+b)^* = \{\varepsilon, a, b, aa, ab, ba, bb, ...\}$

$(r_1 r_2)^* = (ab)^* = \{\varepsilon, ab, abab, ababab, ,....\}$

IV. $(r_1 + r_2)^* \equiv (r_2^* r_1^*)^* \Rightarrow$ True

$(r_1 + r_2)^* = (a+b)^* = \{\varepsilon, a, b, aa, ab, ba, bb, ...\}$

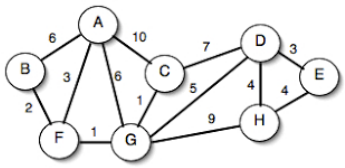
$(r_2^* r_1^*)^* = (b^* a^*)^* = (\{\varepsilon, b, bb, bbb, ...\}\{\varepsilon, a, aa, aaa, ...\})^* = (\{\varepsilon, a, aa, bb, ...\})^* = (a+b)^*$

$= (r_1 + r_2)^*$

D $(r_1 + r_2)^* \equiv (r_2^* r_1^*)^*$

Q.2) Max Marks: 1

Given the following graph shown in the figure below, what is the shortest path reported by Dijkstra's algorithm from vertex B to E_____



Correct Answer

Solution: (11)

Solution: 11

Applying Dijkstra's Algorithm starting from vertex B,

B is popped out initially

A, F are relaxed,

A.cost=6,source=B

F.cost=2,source=B

Now F is popped out from the Queue G and A are relaxed.

G.cost=3, source=F

A.cost=5, source=F

Now G is popped out from the Queue C, H, D are relaxed.

C.cost=4, source=G

H.cost=12, source=G

D.cost=8, source=G

Now C is popped out from the Queue no vertices are relaxed.

Now A is popped out from the Queue no vertices are relaxed.

Now D is popped out from the Queue E is relaxed.

E.cost=11, source=D

Now E is popped out from the Queue no vertices are relaxed.

Now H is popped out from the Queue no vertices are relaxed.

Cost of E from B is 11.

Q.3) Max Marks: 1

Given a block can hold either 3 records or 10 key pointers. A database contains n records, then how many blocks do we need to hold the data file and the dense index.

A **n/3**

B **n/10**

C **n/30**

D **13n/30** Correct Option

Solution: (D)
Solution:
Since the total number of blocks required by data file = n/3
Total number of records needed by pointer = n/10 (key pointer = number of keys)
Therefore, total blocks = n/3 + n/10 = 13n/30

Q.4) Max Marks: 1

Based on the Banking Database Schema given below (primary keys are underlined), answer the following question

- Branch(bname, bcity)
- Customer (cname, city)
- Account (ano, bname, bal)
- Loan (lno, bname, amt)
- Borrower (cname, lno)
- Depositor (cname, ano)

Which of the following statements is TRUE?

A **All account number ano, in depositor relation, have to be distinct.**

B **There must exist one loan number, lno, in borrower relation which is not there in loan relation.**

C **{bname, ano} is a candidate key of the relation account.**

D **Set difference operation, (depositor – borrower) is invalid.** Correct Option

Solution: (D)
Solution:
All the account numbers in depositor relations are not unique, an account can have multiple account holders (joint accounts), hence ano is not the primary key. Each tuple in depositor is uniquely identified by a combination of cname and ano.
Since lno in borrower relation is referencing to lno in loan relation, thus every lno present in borrower has to be in loan relation. This constraint is known as a referential constraint.
Set difference operation depositor borrower is invalid because semantically the attributes of depositor and borrower are different.

Q.5) Max Marks: 1

Consider the table test
(One INTEGER,
Two INTEGER,
PRIMARY KEY (One),
UNIQUE (Two),
CHECK (One > = 1 and < = 10),
CHECK (Two > = 1 and < = 5));
At-most how many tuples this relation can contain?

A **5**

B **10** Correct Option

Solution: (B)
Solution:
They constraint CHECK is used to limit the range that can be placed in the column and since, it is given that "One" is the primary key and it varies from 1 to 10. So, using "One" we can have at-most 10 tuples in the relation. The attribute "Two" is ranged from 1 to 5 but it is a UNIQUE attribute, therefore, it can be NULL. Hence the number of tuples will be 10.

C **15**

D **50**

Q.6) Max Marks: 1

Consider the following items and their given profits and given a knapsack with a capacity of 20 units how many units of item 7 can be added in the optimal filling of the knapsack if we are allowed to take atleast one unit of each item.

Item	Profit	Weight
1	32	4
2	54	7
3	20	5
4	25	12
5	60	6
6	54	9
7	84	12

Correct Answer

Solution: (3)

Solution: 3

Item	Profit	Weight	Profit/Weight
1	32	4	8
2	54	7	7.71
3	20	5	4
4	25	12	2.08
5	60	6	10
6	54	9	6
7	84	12	7

Initially, the profit/weight density is calculated and then the knapsack is filled with the items in order of decreasing profit/weight, given the capacity of the knapsack=20 units of weight.

1. Item 5 is taken and 6 units are taken.
2. Item 1 is taken and 4 units are taken.
3. Item 2 is taken and 7 units are taken. (So far we have used up 17 units and left with only 3 units)
4. Item 7 is taken and only 3 units are taken.
   Therefore the solution is 3 as 3 units of item 7 can be added.

Q.7) Max Marks: 1

Which of the following languages are regular.

A. $L = \{a^n b^l a^k \mid n > 5, l > 3, k \le l\}$

B. Let L_1 and L_2 be regular languages. Then the language $L = \{w \mid w \in L_1, w^R \in L_2\}$

A I Only

B II Only — Correct Option

Solution: (B)

Explanation:

I. $L = \{a^n b^l a^k \mid n > 5, l > 3, k \le l\}$ is not Regular

We need memory element to check the equality between b^l and a^k.

II. $L = \{w \mid w \in L_1, w^R \in L_2\} \Rightarrow \{w \mid w \in L_1\} \cap \{w \mid w^R \in L_2\}$, i.e., $L = L_1 \cap L_2^R$.
Because of the closure of regular languages under intersection and reverse L is regular.

C Both I and II

D Neither I nor II

Q.8) Max Marks: 1

Consider the following DFA

		a	b
→	q_0 F	q_1	q_3
	q_1 F	q_2	q_3
	q_2	q_2	q_5
	q_3	q_4	q_6
	q_4 F	q_2	q_6
	q_5	q_2	q_5
	q_6	q_7	q_3
	q_7 F	q_5	q_6

The number of states in a minimal DFA equivalent to the given DFA

A 7

B 6

C 5

D 4 Correct Option

Solution: (D)

Explanation:

Zero equivalence class π_0 = {q0, q1,q4, q7} {q2,q3, q5, q6}

One equivalence class π_1 = {q0}, {q1,q4,q7} {q2,q5} {q3,q6}

Two equivalence class π_2 = {q0}, {q1,q4,q7} {q2,q5} {q3,q6}

$\pi_1 = \pi_2 \Rightarrow$ Minimum number states in the DFA is 4

Q.9) Max Marks: 1

Given two transactions T1 and T2 working on data item X and Y, where initial values of X and Y are 10 and 20 respectively.
T1: R(x); R(y); y = x + y; W(y)
T2: R(y); R(x); x = x – y; W(x)
Any non-serial interleaving of transactions T1 and T2 allowing concurrent execution will lead to a serializable sequence that is:

A **It will be conflict equivalent to only T2, T1**

B **It will be conflict equivalent to only T1, T2**

C **It is conflict equivalent to both the above mentioned serial schedules.**

D **It does not conflict equivalent to any of the serial schedules.** Correct Option

Solution: (D)
Solution:
Both the serial schedules T1 -> T2 and T2 - > T1, in the very first step one transaction reads the value written by other transactions so any non-serial interleaving will never lead to conflict serializable.

Q.10) Max Marks: 1

Given the following function is to arrange the elements of an unsorted array "a" of size num, in descending order which of the following line is most appropriate to fill in the blanks.

```
1   for (i = 0; i < ( num - 1 ); i++)
2   {
3       for (j = 0; j < num - i - 1; j++)
4       {
5           if (___________)
6           {
7               ___________
8           }
9       }
10  }
```

A **arr[j] < arr[j+1] and swap (arr[j],arr[j+1])** Correct Option

Solution: (A)
Solution:
The above mentioned pseudo code is that of Bubble Sort Algorithm, where each element is compared with its immediately next element if they are in the correct order they are not disturbed otherwise they are swapped, in the question it is given that we want to sort the array in decreasing order, therefore the correct condition is arr[j]<arr[j+1].

B **arr[j] < arr[j+2] and swap (arr[j],arr[j+2])**

C **arr[j] > arr[j+1] and swap (arr[j],arr[j+1])**

D **arr[j] > arr[j+2] and swap (arr[j],arr[j+2])**

Q.11) Max Marks: 1

Suppose that we have an ordered file of 30,000 records on a disk with a block size of 1024 bytes. Records are fixed and are un-spanned of size 100 bytes. Suppose we have created a secondary index on a non-ordering key field of the file that is V = 8 bytes long and block pointer p = 6 bytes long. Find the average number of block access to search a record using indexing.

Correct Answer

Solution: (10)
Solution: 10 block access
Since the key field is 8 bytes long and the block pointer is 6 bytes long, therefore, each index entry Ri = 8 + 6 = 14 bytes.
The blocking factor for the index is bfr = floor (B/Ri) = floor (1024/14) = floor(73.1) = 73
The number of blocks needed for file is b = ceil (r/bfr) = ceil (30,000/10) = 3000 blocks. But in a dense secondary index such as this, the total number of index entries ri is equal to the number of records in the data file. which is 30.000. Therefore. the number of blocks needed for the index is hence bi = ceil(ri/bfr) = ceil(30.000/73) = ceil(410.9) = 411

So, a binary search on this secondary index needs ceil(log2bi) + 1(to search the desired in record in the block) = ceil(log2411) = 9 block access + 1 block access = 10 block access.

Q.12) Max Marks: 1

Given the following function to calculate the factorial of a number, what is the space complexity of the function?

```
int factorial(int n)
{
      int x=1,y=1,f=1;
      for(int i=1;i<=n;i++)
      {
            f=x*i;
            x=f;
      }
      return f;
}
```

A O(n)

B O(1) Correct Option

Solution: (B)
Solution:
The above code makes use of only 4 variables i.e. x,y,f and i which do not depend on the input n, therefore the number of variables are constant and they do not vary with increase of n.

C O(n!)

D None of the above

Q.13) Max Marks: 1

The language generated by the given Context free grammar is
S → 0R0|1R1|ε
R → 00R|01R|10R|11R|ε

A L = {w | w is of even length palindrome}

B L = {w | w is of odd length palindrome}

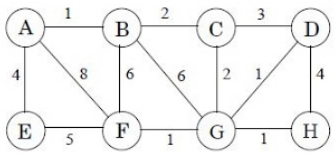
C L = {w | w is of even length and starts and ends with the same symbol} Correct Option

Solution: (C)
Explanation:
S → 0R0|1R1|ε
R → 00R|01R|10R|11R|ε
Strings that are generated by the given grammar is ={ε , 00, 11, 0000,1001, 0000,1111,.....}
Even length strings starts and ends with the same symbol.

D None of these

Q.14) Max Marks: 1

Consider the following graph, what is the cost of the MST for the given graph.



Correct Answer

Solution: (12)
Solution: 12

Applying Kruskal's algorithm
1. Initially adding the edges AB, FG, GH, GD (all of cost 1)
2. Next, adding the edges BC, CG (all of cost 2)
3. CD, DH cannot be added as they form cycles, However edge AE(cost=4) is added and the MST is formed.

Total cost= 4*1 + 2*2 + 4 =12.
Cost of the MST is 12.

Q.15) Max Marks: 1

The language generated by the given context free grammar is
S → 1S|AA
A → 0A1|ε

A $L = \{1^k 0^n 1^n \mid m,n,k \geq 0\}$

B $L = \{1^k 0^n 0^n 1^m 1^m \mid m,n,k \geq 0\}$

C $L = \{1^k 0^n 1^n 0^m 1^m \mid m,n,k \geq 0\}$ Correct Option

Solution: (C)

Explanation:
Given Grammar is
S → 1S|AA
A → 0A1|ε
The Production A → 0A1|ε will generate 0^n1^n
S→ AA, for A we can replace 0A1|ε ⇒ S→ 0^n1^n 0^m1^m | ε
S→ 1S ⇒ 1^K
The language generated by the given grammar is $L = \{1^k 0^n 1^n 0^m 1^m \mid m,n,k \geq 0\}$

D None of these

Q.16) Max Marks: 2

For relation S = (A, B, C, D, E) the following dependencies hold:
A -> BC
E -> A
CD -> E
B -> D
S is decomposed into S1 = (A, B, C, E) and S2 = (B, D). The decomposition is:

A **Lossless decomposition and dependency preserving**

B **Lossy decomposition and dependency preserving**

C **Lossy decomposition and not dependency preserving**

D **Lossless decomposition and not dependency preserving** Correct Option

Solution: (D)
Solution:
For the given decomposition the dependency CD -> E is not preserved but the decomposition is lossless because we can easily merge S1 and S2 as B is the primary key of relation S2 and is common in between S1 and S2. hence, the decomposition is lossless but not dependency preserving.

Q.17) Max Marks: 2

Which of the following statements is/are True
A. The complement of the language $L = \{ww^R : w \in \{a, b\}^*\}$ is context-free.
B. $L = \{ 0^i1^j 0^i \mid i \leq j \}$ is CFL

A **I only** Correct Option

Solution: (A)

Explanation:
I. True
A CFG for L is G = ({S}, {a, b}, S, P) with production P as:
S → a|b|ab|ba|aSa|aSb|bSa|bSb

II. False
$L = \{ 0^i1^j 0^i \mid i \leq j \}$ is Non-CFL ⇒ Requires more than one memory element to check the equality between 0^i1^j and $1^j 0^i$ ($i \leq j$)
(or)
We can prove it by using pumping lemma
Let p be the length provided by the CFG pumping lemma.
Consider the string $0^p1^p0^p$, and suppose it's written as uvxyz with $|vxy| < p$ and $|vy| > 0$.
If either v or y contain a 1, it means vxy can only contain 0s from one of the sides. Then uxz is not in the language because the number of 1s decreases but one side of 0s remains the same.
Otherwise v and x must contain all 0s. Then pumping once, uv^2xy^2z gives a string with too many 0s.

B **II only**

C I and II only

D Neither I nor II

Q.18) Max Marks: 2

Which of the following languages are undecidable

I. L1 = {<M> | M is a Turing machine that rejects all inputs of even length}.
II. L2 = {<M> | M is a Turing machine that halts on an empty input}.
III. L3 = {<M> | there is some input x ∈ {0,1}* such that M accepts x in less than 100 steps}.

A All are undecidable

B I and II only — Correct Option

Solution: (B)

Explanation:

I. L1 is undecidable.
A_{TM} = {<M,w> | M is a TM, w a string, and M accepts w} we know acceptance problem of a Turing Machine is undecidable

L1 = {<M> | M is a Turing machine that rejects all inputs of even length}.
To see this, assume on the contrary that there exists some TM R1 that decides L1, and we use R1 to construct a TM S1 that decides A_{TM}:
S1 = "On input <M, w>:
1. Construct TM M1 that on input x, accept if |x| is odd. If |x| is even, it simulates M on input w. If M accepts w, M1 enters the reject state. If M rejects w, M1 enters the accept state. If M loops, M1 also loops.
2. Run R1 on input <M1>.
3. Accept if R1 accepts, and reject if R1 rejects." Observe that if M accepts w, then M1 is a Turing machine that rejects all inputs of even length. If M rejects or loops on input w, then M1 is a Turing machine that for each input of even length, either loops or accepts.

II. L2 is undecidable.
To see this, assume on the contrary that there exists some TM R2 that decides L2, and we use R2 to construct a TM S2 that decides A_{TM}:
S2 = "On input <M, w>:
1. Construct TM M2 that ignores its input and simulates M on input w and accept (and halt) if M does. If M rejects w, M2 keeps moving right upon reading any input (thereby looping).
2. Run R2 on input <M2>.
3. Accept if R2 accepts, and reject if R2 rejects." Observe that if M accepts w, then M2 is a Turing machine that halts on an empty input. If M rejects or loops on input w, then M2 is a Turing machine that loops on an empty input.

III. L3 is decidable.
First, observe that if <M> ∈ L3, then there exists some string x of length at most 100 such that M accepts x in less than 100 steps. This is because M cannot read beyond the 100th position of its input in less than 100 steps. Therefore, to check whether an input <M> is in L3, it suffices to simulate M over all strings of length at most 100 for at most 99 steps, and accept if M accepts one of these strings, and reject otherwise.

C II and III only

D I and III only

Q.19) Max Marks: 2

Which of the following statements is/are True

I. The union of a regular language with a context free language must be context free
II. The language $\{a^i b^j c^k d^l \mid i=j \text{ and } k=l\}$ is context free
III. The language $\{a^i b^j c^k d^l \mid i+k = j+l\}$ is context free
IV. The class of non-regular languages are closed under complementation

A I and II only

B II and III only

C II, IV only

D I,II,III and IV only — Correct Option

Solution: (D)

Explanation:

I. The union of a regular language with a context free language must be context free **True**
Regular language is a CFL, and CFLs are closed under UNION operation

II. The language $\{a^ib^jc^kd^l \mid i=j \text{ and } k=l\}$ is context free **True**
We can PDA by applying Push Operation on all a's and POP operation on all b's, then apply Push on c's and POP on d's

III. The language $\{a^ib^jc^kd^l \mid i+k=j+l\}$ is context free
We can PDA by applying Push Operation on all a's and c's then apply POP on b's and d's.

IV. The class of non-regular languages are closed under complementation. **True**
If the complement of a language L is regular, then L itself must be regular, therefore the complement of a non-regular language must be non-regular.

Q.20) Max Marks: 2

The time complexity of the below fragment of code is given by

```
int fun(int i)
{
      int count=0;
      while(i>=1)
      {
            i/=2;
            count++;
      }
      return count;
}
int main()
{
      result=0;
      for(i=0;i<=n;)
      {
            result+=(fun(i));
            i=i+(n/2);
      }
      return 0;
}
```

A **O(n)**

B **O(nlogn)**

C $O(n^2)$

D **O(logn)** Correct Option

Solution: (D)

Solution:
The loop inside the main function is executed only 3 times as each time the variable i is incremented by n/2.
In the function fun the inner loop executed
1.0 times when i=0
2. when i=n/2 it is executed log(n/2) times (as each time i inside the while loop is divided by 2 until it reaches 1)
3. When i=n then the while loop is executed log n times.

Total no of times the while loop is executed=logn+log(n/2) =log(n^2/2)=2log(n/2)=O(log n)

Q.21) Max Marks: 2

Consider the following relation and its functional dependencies, where the primary key is underlined

R(U, V, W, X, Y)
UV -> WXY
U -> W
V -> X

Which of the following is correct for the above relation to be in BCNF

A **R1(U, W), R2(V, X), R3(U, V, W, X, Y)**

B **R1(U, W), R2(V, X), R3(U, V, W, Y)**

C **R1(U, W), R2(V, X), R3(U, V, X, Y)**

D **R1(U, W), R2(V, X), R3(U, V, Y)** Correct Option

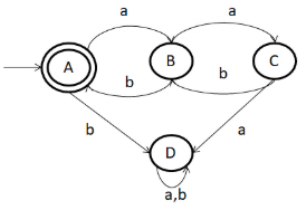
Solution: (D)

Solution:

U determines W uniquely and V determines Y uniquely, while U and V jointly identify Y. Normalizing to 2NF results in all the relations to be normalized to BCNF. So the last relation formed will be R3(U, V, Y)

Q.22) Max Marks: 2

Consider the following deterministic finite automata



The regular expression that represents the given DFA is

A (aabb)*

B (ab)*

C (a(ab)*b)* Correct Option

Solution: (C)

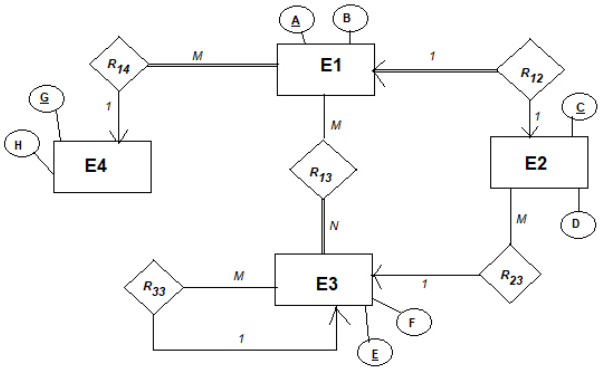
Explanation:

The language that is accepted by the given DFA is
={ε , ab,aabb,abab, ...}
The RE that represents the language is (a(ab)*b)*
Option A. (aabb)* will not generate ab which is a valid string but not generated by the RE
Option B: (ab)* will not generate the string aabb which is also a valid string, but not generated by RE.
Option D: (ab)*(ab)* will not generate the string aabb which is also a valid string

D (ab)*(ab)*

Q.23) Max Marks: 2

Consider the following E-R diagram:



The minimum number of foreign keys required to design a relational model for this E-R diagram is _____

Correct Answer

Solution: (4)

Solution: 4

For entity E1 and E4, E1 will have attribute G as a foreign key to the relation E4 as R14 is a one-to-many relationship with total participation on many sides, therefore they will form 2 separate relations E1 and E4 where G will be the foreign key in E1. Therefore, # of foreign key = 1

For entity E1 and E2, no foreign key is required as relationship R12 is a one-to-one relationship with total participation. Hence, only 1 relation is required. Therefore, # of foreign key = 1

For entity E2 and E3, The attribute E will be the foreign key in E2 (or we should say E1 because E2 is merged with E1 in RDBMS) because R23 is a one-to-many relationship and therefore, attribute E will be placed in E1 as a foreign key. Therefore, # of foreign key = 2

For entity E3 having a self-referential relationship, R33 will have attribute E' as a foreign key in itself as for one-to-many self-referential relationship we don't need a separate relation. Therefore, # of foreign key = 3.

For entity E1 and E3, R13 is a many-to-many relationships with one side total participation. Hence, R13 will not form a separate relation. Therefore, # of foreign key = 3 + 1 = 4

Hence, the total minimum of foreign keys possible is 4.

Q.24) Max Marks: 2

Order the following functions in the decreasing order of their growth

1. $e^{log(n^2+2n+1)}$
2. $(0.9999999)^{n!}$
3. $n^2(logn)^2$
4. 2^{nlogn}

A **4,3,1,2** — Correct Option

Solution: (A)

Solution:

1. $e^{log(n^2+2n+1)}$ can be rewritten as $(n^2+2n+1)*c$ where c is as constant as log and exponential functions are inverse to each other, this will be reduced to O(n^2).

2. $(0.9999999)^{n!}$ is of the form $a^{f(n)}$ where a<1 and f(n) grows the entire expression converges to 0.

3. $n^2(logn)^2$ can be written as $O(n^2(logn)^2)$

4. 2^{nlogn} can be written as $2^n 2^{log\ n}$ =2^n(n) which can be written as O(n2^n).

Among all the fastest-growing is 4 as it is the product of exponential and polynomial function, and the slowest growing is 2 as it converges to 0 as n grows.

We need to compare 1 and 3 to get the correct ordering
1 is O(n^2) whereas 2 is O(n^2 (log n)^2) clearly for as n grows (log n)^2 also grows and therefore 3 grows faster than 1.

The correct ordering is 4>3>1>2

B **4,1,3,2**

C **4,3,2,1**

D **None of the above**

Q.25) Max Marks: 2

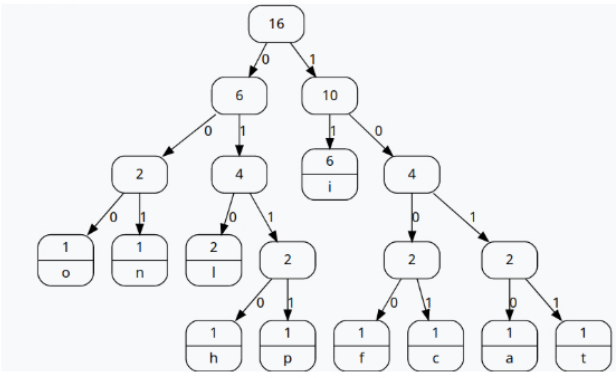
If the compiler of a programming language makes use of Huffman encoding instead of ASCII encoding to store the characters then the no of bytes of memory saved in storing the following string in the memory. Consider one byte is required one character encoded using ASCII encoding. (Ignore the null character '\0' at the end of the string in both the cases)___
"hilipilification"

Correct Answer

Solution: (10)

Solution 10

Using the ASCII encoding the word consists of 16 characters and it consumes 16 bytes of memory.
Following is the Huffman Encoding tree which has the frequency of each of the characters



Length of bits string for each character
I-2
L-3
N-3
O-3
H-4
P-4
F-4
C-4
A-4
T-4
Total no bits required are given by the below table

Character	# of bits	Count	Total
I	2	6	12

I	2	6	12
L	3	2	6
N	3	1	3
O	3	1	3
H	4	1	4
P	4	1	4
F	4	1	4
C	4	1	4
A	4	1	4
T	4	1	4
		Total	48

48 bits correspond to 48/8=6 bytes of memory
No of bytes saved=16-6=10 bytes

close