CS301 - Short Notes

Chapter#1

Question:

(Marks: 2)
If we allow assignment to constants what will happen?
Answer:- (Page 8)
Suppose _a‘ has the value number 3. Now we assigned number 2 the number 3 i.e. all the number 2 will become number 3 and the result of 2 + 2 will become 6. Therefore it is not allowed.

Chapter#2

Question:

Linked memory
(Answer: Page 18)
Linked memory is a memory in which the various cells of memory are not located continuously. In this process, Each cell of the memory not only contains the value of the element but also the information where the next element of the list is residing in the memory. It is not necessary that the next element is at the next location in the memory. It may be anywhere in the memory. We have to keep a track of it.
Thus, in this way, the first element must explicitly have the information about the location of the second element. Similarly, the second element must know where the third element is located and the third should know the position of the fourth element and so on.

Chapter#3

Question:

Drawback of using array to store BST.
(Answer: Page 21)
The drawback of using is the limitations that array being of fixed size can only store a fixed number of elements.
Therefore, no more elements can be stored after the size of the array is reached.

Chapter#4

Question:

Code for removing an element from a linked list data structure if it is implementation by using linked
Memory methods.
Answer: Page 34

```
Void remove() {
If( currentNode != NULL && currentNode != headNode) {
lastCurrentNode->setNext(currentNode->getNext());
delete currentNode;
currentNode = lastCurrentNode->getNext();
size--;
}
};
```

Question:

Write the any two operation of linked list that perform their function in one step.
2Marks
Answer: - (page 38) Addv Removev Findv

Question:

c++ code to add node in doubly link list.
Answer ; (page 40)

```
Class Node {
Public:
Int get() { return object; }; // returns the value of the element
Void set(int object) { this->object = object; }; // set the value of the element
Node* getNext() { return nextNode; }; // get the address of the next node
Void setNext(Node* nextNode) // set the address of the next node
{ this->nextNode = nextNode; };
Node* getPrev() { return prevNode; }; // get the address of the prev node
Void setPrev(Node* prevNode) // set the address of the prev node
{ this->prevNode = prevNode; };
Private:
Int object; // it stores the actual value of the element
Node* nextNode; // this points to the next node
Node* prevNode; // this points to the previous node
};
```

Chapter#7

Question:

Convert the given infix form to postfix form.
A+B/C-D^E-F
Answer:- (Page 66)
AB+C/D-EF -^

Chapter#8

Question:

What is c++ template.
Answer:- (page 75)
A template is a function or class that is written with a generic data type. When a programmer uses this function or
Class the generic data type is replaced with the data type needed to be used in the template function or in the
Template class. We only give the data type of our choice while calling a template function or creating an object of
The template class. The compiler automatically creates a version of that function or class with that specified data
Type.

Chapter#9

Question:

Write the applications of heap.
Answer:- (page 84)
The stack and heap are used in function calls. When we allocate memory dynamically in our programs, it is
Allocated from the heap. The use of priority queue with the help of heap is a major application. The priority queue is
Itself a major data structure, much-used in operating systems. Similarly priority queue data structure is used in
Network devices especially in routers. Heap data structure is also employed in sorting algorithms.

Question:

Two ways for the implementation of Queue data structure.
Answer: - (Page 89, 91) 1. Implementation of queue using linked list
2. Implementation of queue using Array.

Chapter#11&12

Question:

(Marks: 2)
Give the difference between strict and complete binary tree.
Answer:- (Page 121,123) A binary tree said to be a strictly binary tree if every non-leaf node in a binary has non-empty left and right sub trees. If there are left and right sub trees for each node in the binary tree as shown a complete binary tree.

Question:

(Marks: 10)
(a)Write C++ algorithm for Binary Search
Answer:- (Page 125) int isPresent(int *arr, int val, int N) { int low = 0; int high = N – 1; int mid; while (low <= high) { mid = (low + high)/2; if (arr[mid] == val) return 1; // found! Else if (arr[mid] < val) low = mid + 1; else high = mid – 1; } return 0; // not found

Question:

(Marks: 3) How many leaf and non-leaf nodes are present in a complete binary tree if its depth is 7?
Answer:- (Page 125) Leaf Node = 2^d = 2^7 = 128 Non leaf Nodes = 2^d -1 = (2^7) -1 = 128 – 1 = 127

Chapter#17

Question:

Difference between call by value n call by reference (2 marks)
Answer:- (Page 202) In case of call by value, a copy of object is made and placed at the time of function calling in the activation record. By using the references, we are not making the copy but send the address of the variable that function can use the original value.

Chapter#20

Question:

(Marks: 2) Let‘s call the node as a that requires re-balancing. Consider two cases below: 1) An insertion into right subtree of the left child of a. 2) An insertion into left subtree of the right child of a. Which of the following statement is correct about the above two cases. 1) The insertion occurs outside (i.e., right to left or left to right) in the above two cases, single rotation can fix the balance in these two cases. 2) The insertion occurs inside ((i.e., right to left or left to right) in these two cases, single rotation cannot fix the balance in these two cases.
Answer: (page 230)
Insertion occurs on the inside in cases 2 and 3 which a single rotation cannot fix.

Chapter#23

Question:

(Marks: 1) Describe the conditions for second case of deletion in AVL Trees.
Answer:- (Page 266)
This is the case where the parent of the deleted node had a balance of 1 and the node was deleted in the parent‘s left
Sub tree.

Chapter#26

Question:

(Marks: 1) If a Binary Tree has N internal nodes what are the no. of external nodes in it.
Answer:- (Page 304) The No. of external nodes will be N+1

Chapter#28

Question:

(Marks: 2) A complete binary tree can be stored in an array. While storing the tree

in an array we leave the first
Position (0th index) of the array empty. Why?
Answer:- (page 325) In case of having a node with left and right children, stored at position I in the array, the left child will be at position 2i and the right child will be at 2i+1 position. If the value of I is 2, the parent will be at position 2 and the left child will be at position 2i i.e. 4 .The right child will be at position 2i+1 i.e. 5. We have not started the 0th position. It is simply due to the fact if the position is 0, 2i will also become 0. So we will start from the 1 st position, ignoring the 0th.

Chapter#29

Question:

Heap and two types of heap(3)
Answer:- (Page 333) Heap is a data structure of big use and benefit. It is used in priority queue. —The definition of heap is that it is a
Complete binary tree that conforms to the heap order‖. Here heap order means the min and max heap. Max heap: In max heap, each node has a value greater than the value of its left and right child nodes. Moreover, the value of the root will be largest and will become lesser at downward levels. Min heap: in a (min) heap for every node X, the key in the parent is smaller than (or equal to) the key in X. This means that the value in the parent node is less than the value on its children nodes.

Question:

How heap sort works to set a set of data. 2Marks
Answer:- (Page 334) Min heap and max heap are used to sort the data. Min heap convert the data in ascending sorted order and max heap convert the in descending order.

Question:

(Marks: 5) Heapify the elements of the following array (reading from left to right) into a Min Heap and show that
Min Heap contents in the form of array as shown below,
Original array 6 5 3 9 1 2 10 8 –
Answer(page 334)
Min heap
Heapified array 1 5 2 8 6 3 10 9 8 6 5 3 10 2 1 9

Question:

Give two different reasons to explain why the following binary tree is not a heap:
Answer:- (Page 335) Above array is not min heap or max heap because every parent

node has not minimum or maximum value.
In max heap, each parent node has a value greater than the value of its left and right child nodes.
In min heap, each parent node has a value less than the value of its left and right child nodes.

Chapter#30

Question:

(Marks: 3) Consider the following Min Heap apply operation delMin on it and show the resultant Heap,
Answer:- (page 349)
It is clear from the definition of min-heap that it should be in the root of the tree. So finding the minimum is very
Easy. Here node 5 is operational node that has to be deleted.

Question:

(Marks: 3) Explain the process of Deletion in a Min-Heap
Answer:- (Page 350) We have to write a function deleteMin(), which will find and delete the minimum number from the tree. Finding the minimum is easy; it is at the top of the heap.

Chapter#31

Question:

(Marks: 5) Write the code of the perculateDown() function and also comment it.
Answer:- (page 369)

```
Template
Void Heap::percolateDown( int hole )
{
Int child;
eType tmp = array[ hole ];
for( ; hole * 2 <= currentSize; hole = child )
{
Child = hole * 2;
If( child != currentSize && array[child+1] < array[ child ] )
Child++; // right child is smaller
If( array[ child ] < tmp )
Array[ hole ] = array[ child ];
```

Else break;
}
Array[hole] = tmp;
}

Question:

Write the application of binary search tree
Answer: - (page 369) The searching and sorting operations are very common in computer science with the help of binary search tree.

Chapter#33

Question:

What are the properties of equivalence class? 3marks
Answer:- (Page 385) A binary relation R over a set S is called an equivalence relation if it has following Properties 1.Reflexivity: for all element x ∈ S, x R x 2. Symmetry: for all elements x and y, x R y if and only if y R x 3. Transitivity: for all elements x, y and z, if x R y and y R z then x R z

---

Chapter#34

Question:

(Marks: 3)
What is an Equivalent relation? Give any two examples.
Answer : (page 387)
_A binary relation R over a set S is called an equivalence relation if it has following properties‘:
1. Reflexivity: for all element x ȟ S, x R x
2. Symmetry: for all elements x and y, x R y if and only if y R x
3. Transitivity: for all elements x, y and z, if x R y and y R z then x R z
Example; sets of peoples ,electric circuit domain
127 3 7 24 25 20 13 2

Question:

“is a sibling of” set of all human beings is equivalence relation or not? Explain.
Answer: (Page 387)
Yes it is an equivalence relation
First rule is reflexive i.e. for all element x S, x R x. Suppose that x is Haris so Haris R

Haris. This is true because
Everyone is related to each other. Second is Symmetry: for all elements x and y, x R y if and only if y R x. Suppose
That y is Saad. According to the rule, Haris R Saad if and only if Saad R Haris. If two persons are related, the
Relationship is symmetric i.e. if I am cousin of someone so is he. Therefore if Haris is brother of Saad, then Saad is
Certainly the brother of Haris. The family relationship is symmetric. This is not the symmetric in terms of respect but
In terms of relationship. The transitivity is: _for all elements x, y and z. If x R y and y R z, then x R z'. Suppose x is
Haris, y is Saad and z is Ahmed. If Haris —is related to‖ Saad, Saad —is related to‖ Ahmed. We can deduce that Haris
—is related to‖ Ahmed. This is also true in relationships. If you are cousin of someone, the cousin of that person is
Also related to you. He may not be your first cousin but is related to you.

Question:

What is meant by Symmetry in equivalence relations?
Answer:- (Page 387) Symmetry in equivalence relations mean for all elements x and y, x R y if and only if y R x

Question:

What is Disjoint Sets? Explain with an example.
Answer:- (Page 388)
Disjoint sets – Two sets are disjoint if they have no elements in common.
Example:-Suppose there are many people around you. How can we separate those who are related to each other in
This gathering?
We make groups of people who are related to each other. The people in one group will say that they are related to
Each other. Similarly we can have many groups. So every group will say that they are related to each other with
Family relationship and there is no group that is related to any other group. The people in the group will say that
There is not a single person in the other group who is related to them.

Question:

What is Disjoint Sets? Explain with an example.
Answer:- (Page 388)
Disjoint sets – Two sets are disjoint if they have no elements in common.
Example:-Suppose there are many people around you. How can we separate those who are related to each other in

This gathering?
We make groups of people who are related to each other. The people in one group will say that they are related to
Each other. Similarly we can have many groups. So every group will say that they are related to each other with
Family relationship and there is no group that is related to any other group. The people in the group will say that
There is not a single person in the other group who is related to them.

Question:

(Marks: 2) How we can apply Find operation on elements combined through Union operation.
Answer:- (Page 392)
We give an element to the find method and it returns the name of the set.

Chapter#35

Question:

Which is Dynamic Equivalence Problem with an example? (5)
Answer: - (page 395)
We are using sets to store the elements. For this purpose, it is essential to remember which element belongs to
Which set. We know that the elements in a set are unique and a tree is used to represent a set. Each element in
A tree has the same root, so the root can be used to name the set. The item (number) in the root will be unique
As the set has unique values of items. We use this root as the name of set for our convenience. Otherwise,
We can use any name of our choice. So the element in the root (the number) is used as the name of the set.
The find operation will return this name. Due to the presence of many sets, there will be a collection of trees. In
This collection, each tree will be representing one set. If there are ten elements, we will have ten sets initially.
Thus there will be ten trees at the beginning. In general, we have N elements and initially there will be N
Trees. Each tree will have one element. Thus there will be N trees of one node. Now here comes a definition for
This collection of trees, which states that a collection of trees is called a forest.
The trees used for the sets are not necessarily binary trees. That means it is necessary that each node
Should have a maximum of two children nodes. Here a n may have more than two

children nodes.
To execute the union operation in two sets, we merge the two trees of these sets in such a manner that
The root of one tree points to the root of other. So there will be one root, resulting in the merger of the
Trees.
In the find operation, when we call find (x), it helps us to know which set this x belongs to. Internally, we find
This x in a tree in the forest. When this x is found in a tree the find returns the number at root node (the name of the
Set) of that tree.

Question:

(Marks: 5) Consider the following sequence of union commands on the set of elements
{1,2,3,4, 5}:
1 2 3 4 5 union(4,2) union(3,1) union(5,4) union(5,3) Show the result when the unions are performed
Answer: (Page 396 to 398)
4
2
3
51

Question:

Explain and write the code for union and find operation of parent array in disjoint sets.
Answer:- (page 399)
Find (I)
This loop is used to find the parent of an element or the name of the set that contains that element.

```
// traverse to the root (-1)
For(j=I; parent[j] >= 0; j=parent[j])
;
Return j;
```

Union (I, j)
Now let's see the code for the function of union. Here we pass two elements to the function union. The union
Finds the roots of I and j. If I and j are disjoint sets, it will merge them. Following is the code of this function.

```
Root_i = find(i);
Root_j = find(j);
If (root_i != root_j)
```

Parent[root_j] = root_i;
In the code, we see that at first it finds the root of tree in which I exists by the find(i) method and similarly finds
The root of the set containing j by find(j). Then there is a check in if statement to see whether these sets (roots) are
Same or not. If these are not the same, it merges them in such a fashion that the root I is set as the parent of root j.
Thus, in the array where the value of root j exists, the value of root I becomes there.

Question:

(Marks: 3) Which implementation of disjoint set is better with respect space and why?
Answer:- (page 405)
An array that is a single dimensional structure and seems better with respect to space. Suppose we have 1000 set
Members i.e. names of people. If we make a Boolean matrix for 1000 items, its size will be 1000 x 1000. Thus we
Need 1000000 locations for Boolean items, its size will be 1000 x 1000. Thus we need 1000000 locations for
Boolean values. In case of an array the number of location will be 1000. Thus this use of array i.e. tree like structure
Is better then two dimensional arrays in term of space to keep disjoint sets and doing union and find operation.

Chapter#36

Question No: 41 (Marks: 10) Show the result of following sequence of instructions Union(1,2) Union(3,4) Union(3,5) Union(1,7) Union(3,6) Union(8,9) Union(1,8) Union(3,10) Union(3,11) Union(3,12) Union(3,13) Union(14,15) Union(16,17) Union(14,16) Union(1,3) Union(1,14) When the unions are performed by height
Note: You have to show only Final tree, No need to show all steps.
Answer:- (Page 404) Solve it yourself; take help from previous same question that I have solved in page 12 of this file.

Question:

What you conclude from the running time analysis of disjoint sets.
Answer (page 405) We conclude that: union is clearly a constant time operationv Running time of find(i) is proportional to the height of the tree containing node i.v This can be proportional to n in the worst case (but not always).v Goal: Modify union to ensure that heights stay smallv

Question No: 32 (Marks: 3)
Give any three characteristics of Union by Weight method.

Answer:- (Page 408) Following are the salient characteristics of this method: Maintain sizes (number of nodes) of all trees, and duringv union. Make smaller tree, the sub-tree of the larger one.v Implementation: for each root node I, instead of setting parent[i] to -1, set it to -k if tree rooted at I has kv nodes.

Chapter#38

Question:

How we can generate a maze with the help of union
Answer:- (Page 424)
How can we generate maze? The algorithm is as follows:
Randomly remove walls until the entrance and exit cells are in the same set.v Removal of the wall is the same as doing a union operation.v v Do not remove a randomly chosen wall if the cells it separates are already I the same set.

Question:

(Marks: 5) What is Table ADT? Discuss any two implementations of table ADT.
Answer (page 427,428)
The table, an abstract data type, is a collection of rows and columns of information. 6 7 9 8 12 13 6 9 8 7 1 2 13
Implementation of Table
O Unsorted Sequential Array
O Sorted Sequential Array

Question:

What is Table ADT? Discuss any two implementations of table ADT. (Marks: 5)
Answer:- (Page 427, 428)
The table, an abstract data type, is a collection of rows and columns of information.
Implementation of Table
O Unsorted Sequential Array
O Sorted Sequential Array

Question:

What is an ADT?
Answer:- (Page 428)
The table, an abstract data type, is a collection of rows and columns of information.

Question:

(Marks: 2)

Give the operation names that we can perform on Table abstract data type.
Answer:- (page 429) Insert Find Remove

Chapter#39

Question:

Consider the following array of values; show how the binary search algorithm would find the value -5.
Clearly show/explain your work; that is, show the values of start, end, etc. for each step of the algorithm.
Answer : (page 437) Value < mid Value = -5 Mid=(0+8)/2=4 Low=0 High=mid -1=4-1 -15 -5 0 7 13 16 27 30 42 0 1 2 3 4 5 6 7 8 low high mid Starting from the low which is -15.now compare the -5 form the next index which is at position 1. -15 -5 0 7 13 16 27 30 42 0 1 2 3 4 5 6 7 8 low high mid -15 -5 0 7 13 16 27 30 42 0 1 2 3 4 5 6 7 8 low high mid

---

Chapter#40

Question:

How we can implement Table ADT using Linked?
Answer:- (Page 441) List linked list is one choice to implement table abstract data type. For unsorted elements, the insertion at front operation will take constant time. But if the data inside the list is kept in sorted order then to insert a new element in the list, the entire linked list is traversed through to find the appropriate position for the element.

Question:

Important characteristics of skip list.
Answer ; (page 442)
Search and update require linear time
Fast Searching of Sorted Chain
Provide alternative to BST (binary search trees) and related tree structures.
Balancing can be expensive.
Relatively recent data structure: Bill Pugh proposed it in 1990

Question:

What is a skip List?
Answer:- (Page 446)

A skip list for a set S of distinct (key, element) items is a series of lists S0, S1 , … , Sh such that
• Each list Si contains the special keys +∞ and -∞
• List S0 contains the keys of S in non-decreasing order
•Each list is a subsequence of the previous one, i.e.,
S0 ⊇ S1 ⊇ … ⊇ Sh
•List Sh contains only the two special keys

Question:

How can we search an element in skip list? 2Marks
Answer:- (Page 447) We search for a key x in the following fashion:
We start at the first position of the top list
• At the current position p, we compare x with y key(after(p))
• x = y: we return element(after(p))
• x > y: we —scan forward‖
• x < y: we —drop down‖
• If we try to drop down past the bottom list, we return NO_SUCH_KEY

Chapter#41

Question:

Which three basic operations are performed on AVL trees (mark 3)
Answer:- (page 457) Insertv Deletev Searchv

Chapter#42

Question:

Write one example of hashing?
Answer: (page 459) Hashing can be used to store the data. Suppose we have a list of fruits. The names of Fruits are in string. The key is the name of the fruit. We will pass it to the hash Function to get the hash key.
HashCode (“apple”) = 5
We pass it to the string —apple‖. Resultantly, it returns a number 5.

Question:

(Marks: 5)
Explain the following terms:
1. Collision

2. Linear Probing
3. Quadratic Probing
Answer:- (Page 464-470)
Collision:
When two values hash to the same array location, this is called a collision..
Linear Probing When there is a collision, some other location in the array is found. This is known as linear probing. In linear probing, at the time of collisions, we add one to the index and check that location. If it is also not empty, we add 2 and check that position. Suppose we keep on incrementing the array index and reach at the end of the table. We were unable to find the space and reached the last location of the array.

Chapter#44

Question:

(Marks: 2)
What are different applications of Hashing?
Answer (page 474) Compilers use hash tables to keep track of declared variables (symbol table).v A hash table can be used for on-line spelling checkersv — if misspelling detection (rather than correction) is important, an entire dictionary can be hashed and words checked in constant time. Game playing programs use hash tables to store seen positions, thereby saving computation time if thev position is encountered again.

Question:

(Marks: 3) When Hashing is NOT suitable?
Answer: - (Page 475) Hash tables are not so good if there are many insertions and deletions, or if table traversals are needed — in this case, AVL trees are better.

Question:

(Marks: 5)
Write down the C++ code from Selection Sort Algorithm.
Answer:- (Page 480) void selectionSort(int *arr, int N) { int posmin, count, tmp ; for (count=0;count

Question:

Write down the C++ code to implement insertion sort algorithm.
Answer:- (Page 483)
Following is the code of the insertion sort in C++.
Void insertionSort(int *arr, int N)

```
{
Int pos, count, val;
For(count=1; count < N; count++)
{
Val = arr[count];
For(pos=count-1; pos >= 0; pos--)
If (arr[pos] > val)
Arr[pos+1]=arr[pos];
Else break;
Arr[pos+1] = val;
}
```

Chapter#45

Question:

Name of two divide and conquer algorithm (2 marks)
Answer:- (Page 488)
Merge Sortv
Quick Sortv
Heap Sortv
These three algorithms fall under _divide and conquer category‘.

==========

Question No: 36 (Marks: 3)
Suppose we are sorting an array of eight integers using a some quadratic sorting algorithm. After four
Iterations of the algorithm‟s main loop, the array elements are ordered as shown here:
2 4 5 7 8 1 3 6
Which statement is correct? (Note: Our selection sort picks largest items first.)
A. The algorithm might be either selection sort or insertion sort.
B. The algorithm might be selection sort, but it is not insertion sort.
C. The algorithm is not selection sort, but it might be insertion sort.
D. The algorithm is neither selection sort nor insertion sort.
E. None of these.
Answer:-
C. The algorithm is not selection sort, but it might be insertion sort. (Correct)

Question No: 38 (Marks: 5) If we insert a new element into an AVL tree of height 4, is one rotation sufficient to re-establish balance? Justify your answer.

Answer:-
No, one rotation is not always sufficient to re-establish balance. For example, consider the insertion of the shaded
Node in the following AVL tree:
Though the original tree was balanced, more than one rotation is needed to restore balance following the insertion.
This can be seen by an exhaustive enumeration of the rotation possibilities.
The problem asks for a tree of height 4, so we can extend the above example into a larger tree:

Question:

For smaller list linear insertion sort is perform well but for large list Quick sort suitable to apply/justify
Why????????
Answer:- Since for smaller lists number of comparisons and interchange (if needed) decreases radically in insertion sort than quick sort. For larger lists, opposite occurs in both sorting techniques.

Question:

Consider the following Binary tree and Delete the key 29, 35, 75 and 60 only. Show the
Resultant tree after deleting these keys. (5)
Answer:-

Question:

Worst case of selection sort
Answer: - Click here for detail The worst case occurs if the array is already sorted in descending order. Nonetheless, the time require by selection sort algorithm is not very sensitive to the original order of the array to be sorted.

Question:

Height of a tree is 5 find sum of heights (2)
Answer:- Height = 5 Nodes = 2^h = 2^5 =32 Sum of Nodes = N-(h+1)= 32-(5+1) = 32-6 = 26

Question No: 5
Here is an array with exactly 15 elements:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15.
Suppose that we are doing a binary search for an element. Indicate any elements that will be found by
Examining two or fewer numbers from the array. (Marks: 5)
Answer:- Click here for detail

If there is an array of 15 elements, {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}
It wants to know which elements can be discovered by examining two or fewer elements. I want assume the
Divide and conquer aspect of a binary search, so that leads me to think these elements will be found
1, 2, 14, 15, 7, 3 rather; 1, 2, 3, 4, 6, 8, 14, 15 1 and 15 for heads and tails 2, and 14 for next elements (following or preceding) ½ of 15 is either 6, or 8, from 7.5 and ½ of 6 = 3 ½ 8 = 4

Question:

How do we carry out degeneration of Complete Binary Tree?
Answer: Click here for detail Degeneration of the (original) binary tree is mainly reflected by degeneration of the internal infix trees and less by degeneration of the external height of the PPbin tree.

Question:

Consider the following max heap. Add node 24 in it and show the resultant Heap.
(Marks: 3)
Answer:-

Question:

Here is an array of 10 integers.
5 3 8 1 7 0 2 6 4.
Draw this array after the first iteration of the large loop in a selection sort (Sorting from smallest to largest)
Answer:-
5 3 8 9 1 7 0 2 6 4
1st step 0 3 8 9 1 7 5 2 6 4
2nd step 0 1 8 9 3 7 5 2 6 4
3rd step 0 1 2 9 3 7 5 8 6 4
4th step 0 1 2 3 9 7 5 8 6 4
5th step 0 1 2 3 4 7 5 8 6 9
6th step 0 1 2 3 4 5 7 8 6 9
7th step 0 1 2 3 4 5 6 8 7 9
8th step 0 1 2 3 4 5 6 7 8 9

Question:

Write different type of hashing. 2 Marks
Answer: - Click here for detail
Data structures that employ hashing methods to store keys (and information possibly associated to it) are denoted
As hashing types.

Question:

Give the table name that that we can on table abstract data type. 3 marks

Question:

77, 33,44,11,88,22,66,55 sort the following array using insertion algorithm. 5 Marks
Answer: - See the same solution above

Question:

Suppose we have the following representation for a complete Binary Search Tree, tell the Left and Right
Child nodes and Parent node of the node D (Marks: 5) A B C D E F G H I J K L M N O P Q R S T … 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 …
Answer:- parent: B using formula(i/2) left child:-H 2i=2*4=8=H Right child:-I 2i+1=8+1=9=I

Question:

“For smaller lists, linear insertion sort performs well, but for larger lists, quick sort is suitable to apply.”
Justify why?
Answer:- Since for smaller lists number of comparisons and interchange (if needed) decreases radically in insertion sort than quick sort. For larger lists, opposite occurs in both sorting techniques.

Question:

Consider the following Threaded Binary Tree, You have to give the values that should be in the four variables given below, for the node 37 1. LTH (Left flag) 2. RTH (Right flag) 3. Left node pointer (->L) 4. Right node pointer (->R)

Question:

Give the effect of sorted data on Binary Search.
Answer:- http://www.cprogramming.com/tutorial/lesson18.html The binary tree is a fundamental data structure used in computer science. The binary tree is a useful data structure for rapidly storing sorted data and rapidly retrieving stored data.

Question:

Here is an array of ten integers:
5 3 8 9 1 7 0 2 6 4
Draw this array after the FIRST iteration of the large loop in an insertion sort (sorting from smallest to

Largest). This iteration has shifted at least one item in the array!
Answer:-
5 3 8 9 1 7 0 2 6 4
1 st step 3 5 8 9 1 7 0 2 6 4
2nd step 1 3 5 8 9 7 0 2 6 4
3rd step 1 3 5 7 8 9 0 2 6 4
4th step 0 1 3 5 7 8 9 2 6 4
5th step 0 1 2 3 5 7 8 9 6 4
6th step 0 1 2 3 5 6 7 8 9 4
7th step 0 1 2 3 4 5 6 7 8 9

Question:

Give your comment on the statement that heap uses least memory in array representation of binary
Trees. Justify your answer in either case.
Answer:- A heap is a complete binary tree, so it is easy to be implemented using an array representation It is a binary tree with the following properties:- Property 1: it is a complete binary treev Property 2: the value stored at a node is greater or equal to the values stored at the childrenv From property 2 the largest value of the heap is always stored at the rootv

Quadratic Probing In the quadratic probing when a collision happens we try to find the empty location at index + 1^2. If it is filled then we add 2^2 and so on. Quadratic probing uses different formula:
1 . Use F(i) = i2 (square of i) to resolve collisions 2. If hash function resolves to H and a search in cell H is inconclusive, try H + 1 2, H + 22, H + 32

Question:

(Marks: 2) Convert this tree representation of a heap into the corresponding array representation
Answer:-

100 19 36 17 3 25 1 2 7

Question:

(Marks: 2)
Give your comments on the statement:
“Efficiently developed data structures decrease programming effort”
Answer:- Click here for detail
Reduces programming effort by providing useful data structures and algorithms so you don’t have to write them yourself.
Increases performance by providing high-performance implementations of useful data structures and algorithms. Because the various implementations of each interface are

interchangeable, programs can be easily tuned by switching implementations.

Question:

(Marks: 5) Here is an array of ten integers: 5 3 8 9 1 7 0 2 6 4 Show the first three merging steps for Merge sort on this array.
Answer:-
5 3 8 9 1 7 0 2 6 4
1st step 5 3 8 9 1 7 0 2 6 4
2nd step 3 5 8 9 1 7 0 2 4 6
3rd step 3 5 8 9 1 7 0 2 4 6
4th step 1 3 8 5 9 2 6 7 0 4
5th step 1 3 8 5 9 2 6 7 0 4
6th step 1 2 3 6 7 8 0 5 4 9

Question:

(Marks: 2) Represent the following Binary tree using array representation.
Answer:-
A B C D E

Question No: 35 (Marks: 5)
Suppose the following numbers are stored in an array „BS‟:
32,51,27,85,66,23,13,57
We apply Bubble sort to array „BS‟. Show step by step sorting. You have to write only two passes that is
Passing 1 and pass 2.
Hint; while concern step by step sorting there are 6 passes involved to sort array nums. You have to show
All passes
Answer:-
Pass one:-
32 51 27 85 66 23 13 57
Step1 32 51 27 85 66 23 13 57
Step2 32 27 51 85 66 23 13 57
Step3 32 27 51 66 85 23 13 57
Step4 32 27 51 66 23 85 13 57
Step5 32 27 51 66 23 13 85 57
Step6 32 27 51 66 23 13 57 85
Pass two:-
32 27 51 66 23 13 57 85
Step1 32 27 51 66 23 13 57 85
Step2 27 32 51 66 23 13 57 85
Step3 32 27 66 23 66 13 57 85
Step4 32 27 66 23 13 66 57 85
Step5 32 27 66 23 13 57 66 85

Step6 32 27 66 23 13 57 66 85