Q.1 c. What is the bucket size, when the overlapping and collision occur at the same time?

Answer:

Bucket size: one. If there is only one entry possible in the bucket, when the collision occurs, there is no way to accommodate the colliding value. This results in the overlapping of values.

d. There are 8, 15, 13 and 14 nodes in 4 different trees. Which one of them can form a full binary tree?

Answer:

The answer is the tree with 15 nodes. In general, there are 2^n-1 nodes in a full binary tree. Thus, the correct answer is 15.

e. What pointer type is used to implement the heterogeneous linked list in C?

Answer:

Void pointer. The heterogeneous linked list contains different data types in it's nodes and we need a link, pointer, to connect them. Since we can't use ordinary pointers for this, we use the void pointer. Void pointer is a generic pointer type, and capable of storing pointer to any type.

f. Does the minimum spanning tree of a graph give the shortest distance between any 2 specified nodes? Explain.

Answer:

No. The Minimal spanning tree assures that the total weight of the tree is kept at its minimum. But it doesn't mean that the distance between any two nodes involved in the minimum-spanning tree is minimum.

g. What is the difference between B-tree and B+ tree?

Answer:

In a B- tree you can store both keys and data in the internal/leaf nodes. But in a B+ tree you have to store the data in the leaf nodes only.

Q.2 a. Differentiate between NULL and VOID.

Answer:

Null is actually a value, whereas Void is a data type identifier. A variable that is given a Null value simply indicates an empty value. Void is used to identify pointers as having no initial size.

b. How can you dynamically allocate a multidimensional array? Write C code.

Answer:

```
#include <stdlib.h>
int **array1 = malloc(nrows * sizeof(int *));
for(i = 0; i < nrows; i++)
array1[i] = malloc(ncolumns * sizeof(int));
```

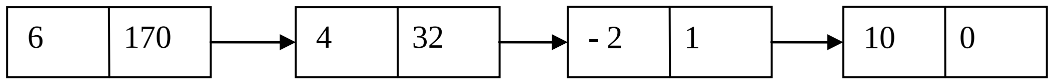
Q.3 a. Write a C program to create a copy of a linked list.

Answer:

```
copy_linked_lists(struct node *q, struct node **s)
{
if(q!=NULL)
{
*s=malloc(sizeof(struct node));
(*s)->data=q->data;
(*s)->link=NULL;
copy_linked_list(q->link, &((*s)->link));
}
}
```

b. How a polynomial such as $6x^{170}+4x^{32}-2x+10$ can be represented by linked list? Write an algorithm that reads such a polynomial.

Answer:



Q.4 b. Write a function to compute the maximum depth in a tree?

Answer:

```
int maxDepth(struct node* node)
{
if (node==NULL)
{
return(0);
}
else
{
int leftDepth = maxDepth(node->left);
int rightDepth = maxDepth(node->right);
if (leftDepth > rightDepth) return(leftDepth+1);
else return(rightDepth+1);
}
}
```

Q.5 a. Find the binary tree whose inorder and preorder traversals is given below:

inorder = g d h b e i a f j c
preorder = a b d g h e i c f j

Answer:

Scan the preorder left to right using the inorder sequence to separate left and right subtrees. For example, "a" is the root of the tree; "gdhbei" are in the left subtree; "fjc" are in the right subtree. "b" is the next root; "gdh" are in the left subtree; "ei" are in the right subtree. "d" is the next root; "g" is in the left subtree; "h" is in the right subtree.

b. The keys 12, 18, 13, 2, 3, 23, 5 and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function h(k) = k mod 10 and linear probing. What is resultant hash table?

Answer:

The resultant hash table:

0	
1	
2	12
3	13
4	2
5	3
6	23
7	5
8	18
9	15

Q.6 b. Show the steps of Huffman's algorithm for the following set of frequencies

f: 5 e: 9 c: 12 b: 13 d: 16 a: 45

Answer:

Page 341 Algorithms by Cormen

Q.7 a. The transpose of a directed graph G = (V, E) is the graph $G^T = (V, E^T)$, where $E^T = \{(v, u) \in V \times V : (u, v) \in E\}$. Thus, G^T is G with all its edges reversed. Describe efficient algorithms for computing G^T from G, for both the adjacency list and adjacency matrix representations of G. Analyze the running times of your algorithms.

Answer:

The transpose of a directed graph G^T can be computed as follows: If the graph is represented by an adjacency-matrix A simply compute the transpose of the matrix A^T in time $O(V^2)$. If the graph is represented by an adjacency-list a single scan through this list is sufficient to construct the transpose. The time used is O(E + V).

b. What do you mean by buddy system memory allocation? What are its drawbacks?

Answer:

Free memory is maintained in linked lists, each of equal sized blocks. Any such block is of size 2k. When some memory is required by a process, the block size of next higher order is chosen, and broken into two. The two such pieces differ in address only in their kth bit. Such pieces are called buddies. When any used block is freed, the OS checks to see if its buddy is also free. If so, it is rejoined, and put into the original free-block linked-list.

Text Books

1. B.W. Kernighan and D.M. Ritchie, “The C Programming Language”, Prentice Hall of India, 1989

2. E. Horowitz, S. Sahai and S Anderson, “Fundamentals of Data Structures in C” Silicon Press, 2007