Q2 (a) Compare the advantages and disadvantages when a list of numbers are represented using
(i) an array (ii) a linked list

Answer

```
typedef struct node { int value; struct node *link;} Node;
Node move(Node *head)
{
        if ((head == 0) || (head->link == 0)) return head;
        node *p, *q;
        q = 0; p = head;
        while (p->next 1= 0l)
        {       q = p;  p = p->link;}
        q->next = 0;
        p->next = head;
        head = p;
        return head;
}
```

Q2 (b) Consider the following recursive C function that takes two arguments.

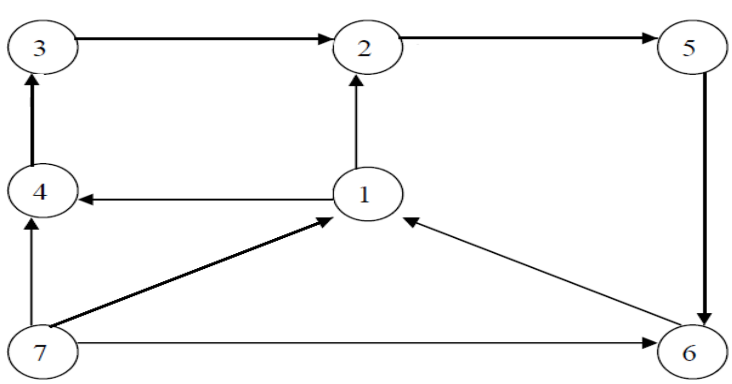
```
unsigned int fun (unsigned int n, unsigned int r)
    {   if (n > 0) return ((n%r) + fun (n/r, r));
    else return 0;
     }
```

What is the return value of the function when it is called as fun (345, 10)?

Answer

fun(345, 10) = 5 + fun(34, 10) = 5 + 4 + fun(3, 10) = 5 + 4 + 3 + fun(0, 10) = 12

Q2(c) Consider the directed graph given:
Write down the adjacency matrix and adjacency list for the graph.
Compare the memory space requirement for the two representations for the graph.

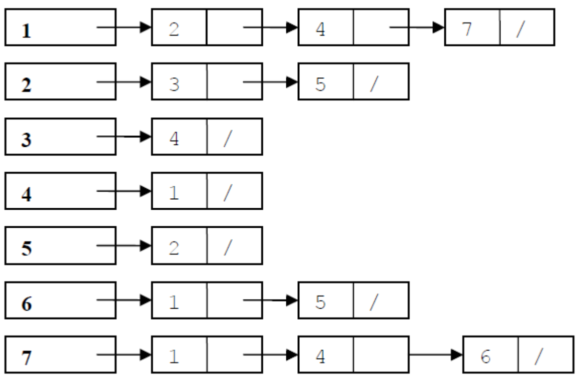


Answer

Matrix

	1	2	3	4	5	6	7
1	0	1	0	1	0	0	1
2	0	0	1	0	1	0	0
3	0	0	0	1	0	0	0
4	1	0	0	0	0	0	0
5	0	1	0	0	0	0	0
6	1	0	0	0	1	0	0
7	1	0	0	1	0	1	0

List



If the integer value consumes 4 bytes:

Matrix = 49 * 4 = 196 bytes

List = 20*(4+4) = 160 bytes

Q3 (a) Arrange the following functions in the increasing order of asymptotic complexity. Justify your answer for n=1024.

$f_1(n) = 2^n$ **$f_2(n) = n^{3/2}$**

$f_3(n) = n \log_2 n$ **$f_4(n) = n^{\log_2 n}$**

Answer

$$n \log n \leq n^{3/2} \leq n^{\log n} \leq 2^n$$

$$\text{Let} \quad n = 1024$$

$$f_1(n) = 2^{1024}$$

$$f_2(n) = 2^{15}$$

$$f_3(n) = 10 \times 2^{10}$$

$$f_4(n) = 1024^{10} = 2^{100}$$

Q3 (b) What is Tower of Hanoi puzzle? Write the recursive algorithm for the same. Derive the recurrence relation capturing the optimal execution time of the puzzle with n discs.

Answer

Definition on page 96 of the Text Book.

TOH (A, B, C, n)
{

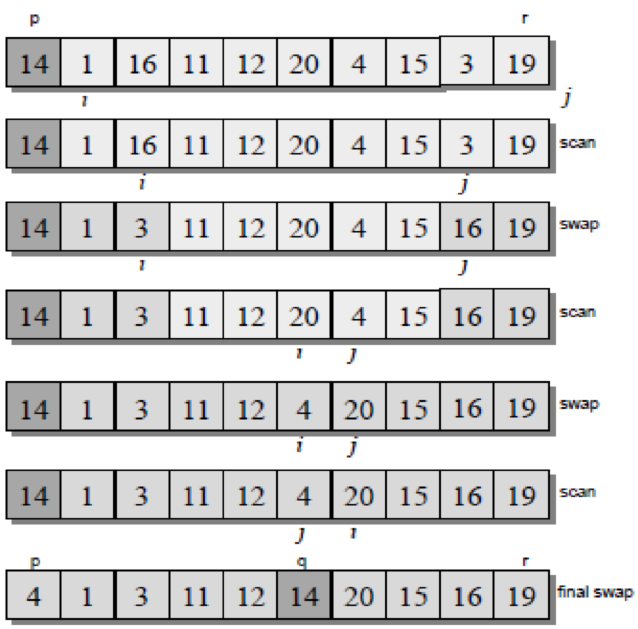
TOH (A, C, B, n – 1); (T (n - 1))
A → C 1
TOH (B, C, A, n-1) T(n-1)
}
Hence T (n) = T (n – 1) + 1 + T (n – 1)
= 2 T (n – 1) + 1

Q4 (a) Write the algorithm for selection sort and derive its time complexity.

Answer Page Number 123 of the Text Book

Q4 (b) Consider an array of integers [14 1 16 11 12 20 4 15 3 19]. Illustrate the operation of partition of Quicksort on this array. Indicate where the pivot element lyes when the algorithm terminates.

Answer



Q4 (c) Use Strassen's matrix multiplication algorithm to multiply

$$\mathbf{X} = \begin{bmatrix} 3 & 2 \\ 4 & 8 \end{bmatrix} \text{ and } \mathbf{Y} = \begin{bmatrix} 1 & 5 \\ 9 & 6 \end{bmatrix}$$

Answer
Let Z = X . Y and partition each matrix into four sub-matrices. Accordingly, A = [3], B = [2], C = [4], D = [8], E = [1], F = [5], G = [9] and H = [6], where,

$$\mathbf{Z} = \begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{K} & \mathbf{L} \end{bmatrix}, \mathbf{X} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \text{ and } \mathbf{Y} = \begin{bmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{bmatrix}$$

Applying Strassen's algorithm, compute the following products:
(i) S1 = A . (F- H) = [3] . ([5] - [6]) = [-3].
(ii) S2 = (A + B) . H = ([3] + [2]) . [6] = [30].
(iii) S3 = (C + D) . E = ([4] + [8]) . [1] = [12].
(iv) S4 = D . (G - E) = [8] . ([9] - [1]) = [64].
(v) S5 = (A + D) . (E +H) = ([3] + [8]) . ([1] + [6]) = [77].
(vi) S6 = (B - D). (G+H) = ([2] - [8]) . ([9] + [6]) = [-90].
(vii) S7 = (A - C) . (E + F) = ([3] - [4]) . ([1] + [5]) = [-6].

Compute **Z** as follows:
(i) I = S5 + S6 + S4 - S2 = 21
(ii) J = S1 + S2 = 27
(iii) K = S3 + S4 = 76
(iv) L = S1 - S7 - S3 + S5 = 68

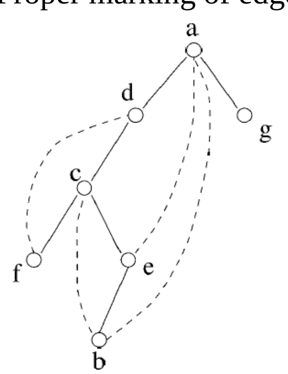
⇨

Q5 (b) Explain Johnson-Trotter algorithm, generate all permutations of 1, 2, 3 and 4.

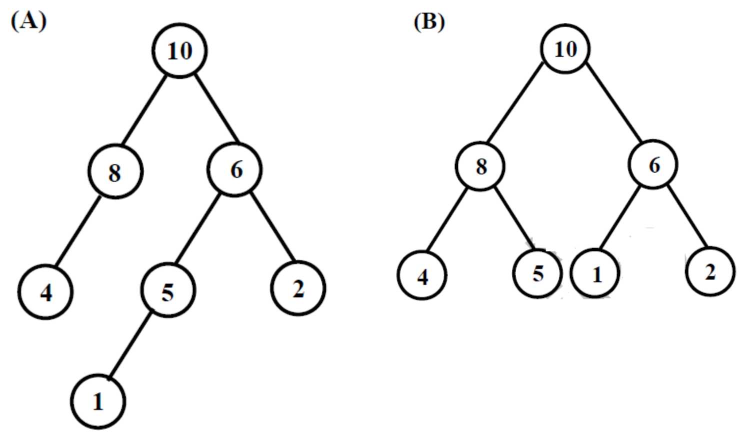
Answer

DFS tree –

Proper marking of edges -



Q6 (a) Define max-heap. Are the trees given below max-heaps? Justify your answer.



Answer

Definition of max-heap is available on Page Number 241 of the TextBook

The structure of a heap is near-complete binary tree. All internal nodes except possibly in last two levels must have two children. Tree in figure **A** does not have this property. Tree in Figure B is a max-heap.

Q7 (a) Write the pseudo code for Floyd's algorithm and explain.

Answer

Floyd algorithm

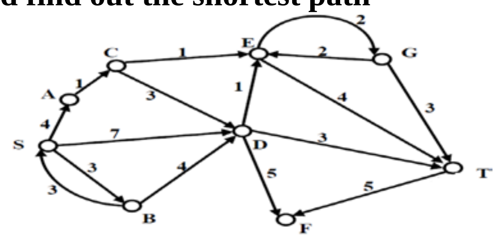
$$D^{(0)} = \begin{bmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & \infty & \infty \\ \infty & 2 & 0 & \infty & \infty & -8 \\ -4 & \infty & \infty & 0 & 3 & \infty \\ \infty & 7 & \infty & \infty & 0 & \infty \\ \infty & 5 & 10 & \infty & \infty & 0 \end{bmatrix}$$

$$D^{(1)} = \begin{bmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & 0 & \infty \\ \infty & 2 & 0 & \infty & \infty & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ \infty & 7 & \infty & \infty & 0 & \infty \\ \infty & 5 & 10 & \infty & \infty & 0 \end{bmatrix}$$

$$D^{(3)} = D^{(2)} = \begin{bmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & 0 & \infty \\ 3 & 2 & 0 & 4 & 2 & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ 8 & 7 & \infty & 9 & 0 & \infty \\ 6 & 5 & 10 & 7 & 5 & 0 \end{bmatrix} \qquad D^{(5)} = \begin{bmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ 0 & 2 & 0 & 4 & -1 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{bmatrix}$$

$$D^{(4)} = \begin{bmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ 0 & 2 & 0 & 4 & -1 & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{bmatrix} \qquad D^{(6)} = \begin{bmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ -5 & -3 & 0 & -1 & -6 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{bmatrix}$$

Q7 (b) Consider the directed graph shown in the figure below. There are multiple shortest paths between vertices S and T. Apply Dijkstra's algorithm and find out the shortest path



Answer

Let Q be the set of vertices for which shortest path distance has not been computed. Let W be the set of vertices for which shortest path distance has not been computed. Initially, Q = {S, A, B, C, D, E, F, G, T}, W = ϕ d[S] = 0, d[A] = ∞, d[B] = ∞, ... , d[T] = ∞

vertex from Q with minimum d[u] value	Q	d		
S	{A, B, C, D, E, F, G, T}	d[S] = 0, d[A] = 4, d[B] = 3, d[C] = ∞, d[D] = 7, d[E] = ∞ - - -, d[T] = ∞	P[A] = S, P[B] = S, P[C] = 1, P[D] = S, P[E] = 1 - - - , P[T] = 1	W = {S}
B	{A, C, D, E, F, G, T}	d[S] = 0, d[A] = 4, d[B] = 3, d[C] = ∞, d[D] = 7, d[E] = ∞ - - -, d[T] = ∞	P[A] = S, P[B] = S, P[C] = 1, P[D] = S, P[E] = 1 - - - , P[T] = 1	{S, B}
A	{C, D, E, F, G, T}	d[S] = 0, d[A] = 4, d[B] = 3, d[C] = 5, d[D] = 7, d[E] = ∞ -	P[A] = S, P[B] = S, P[C] = A, P[D] = S, P[E] = 1 - - - , P[T] =	W = {S, B, A}

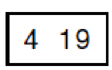
		- -, d[T] = ∞	1	
C	{D, E, F, G, T}	d[S] = 0, d[A] = 4, d[B] = 3, d[C] = 5, d[D] = 7, d[E] = 6, - - -, d[T] = ∞	P[A] = S, P[B] = S, P[C] = A, P[D] = S, P[E] = C, - - - , P[T] = 1	W = {S, B, A, C}
E	{D, F, G, T}	d[S] = 0, d[A] = 4, d[B] = 3, d[C] = 5, d[D] = 7, d[E] = 6, d[F] = ∞, d[G] = 8, d[T] = 10	P[A] = S, P[B] = S, P[C] = A, P[D] = S, P[E] = C, P[F] = 1, P[G] = E, P[T] = E	W = {S, B, A, C, E}

We observe that P[T] = E, P[E] = C, P[C] = A, P[A] = S, So the shortest path from S to T is SACET

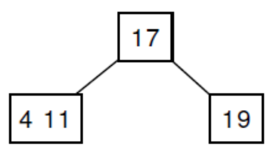
Q8 (a) Show the result of inserting the keys 4,19, 17, 11, 3, 12, 8, 20, 22, 23, 13, 18, 14, 16, 1, 2, 24, 25, 26, 5 in order to an empty B-Tree of degree 3. Only draw the configurations of the tree just before some node must split, and also draw the final configuration.

Answer

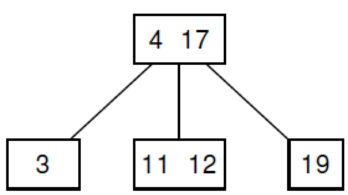
4, 19



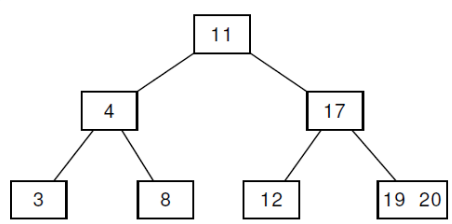
17, 11



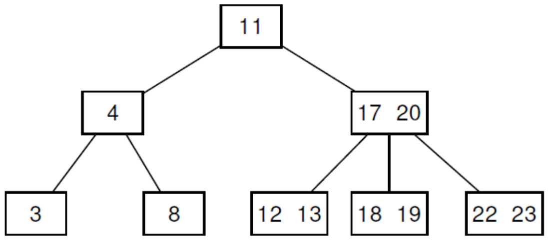
3, 12



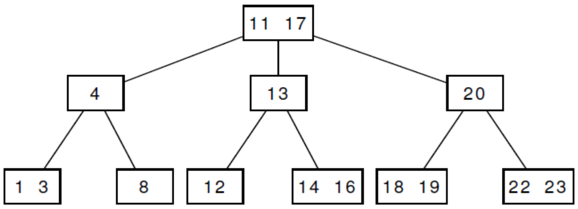
8, 20



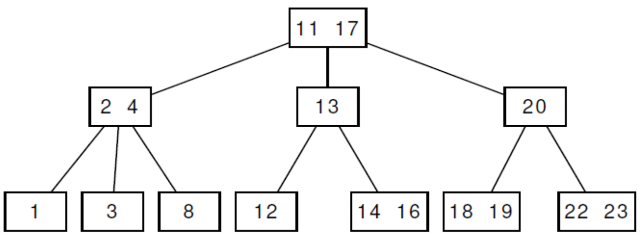
22, 23, 13, 18



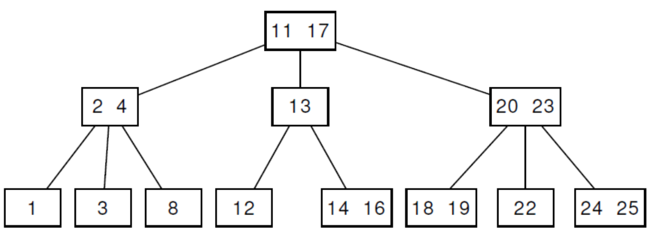
14, 16, 1



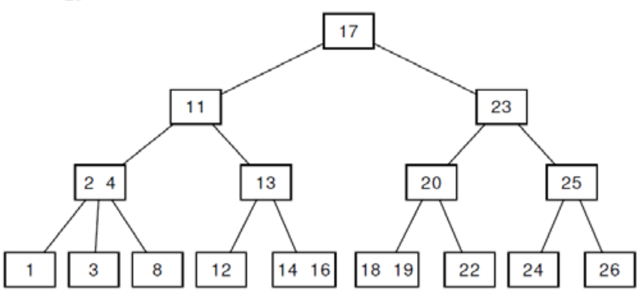
2



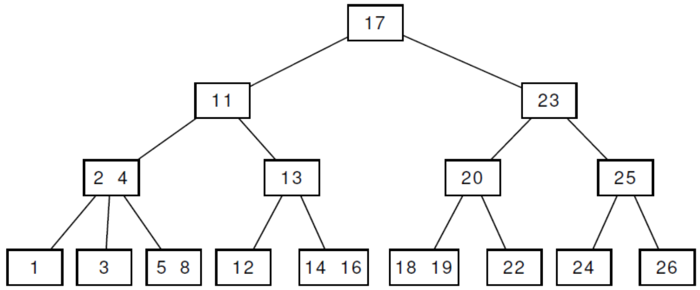
24, 25



26



5

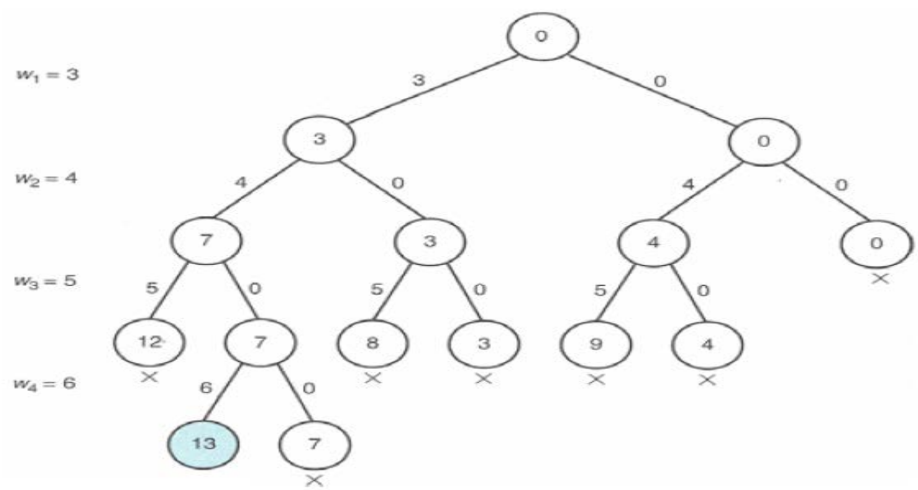


Q8 (b) Define NP-complete decision problem. Consider the example of Hamiltonian circuit and explain how closely related decision problems are polynomially reducible.

Answer Page Number 375 of Text Book

Q9 (a) Define sum of subset problem. Apply backtracking to solve the following instance of sum of subset problem: w= (3, 4, 5, 6} and d = 13. Briefly explain the method using a state-space tree.

Answer



Q9 (b) What are commonalities and differences between backtracking and branch and bound algorithms

Answer

<u>Commonalities</u>:

i) Both strategies can be considered as an improvement over exhaustive search. Unlike exhaustive search, they construct candidate solutions one component at a time and evaluate the partially constructed solution: if no potential values of the remaining components can lead to solution, the remaining components are not generated at all.

ii) They are based on the construction of a state-space tree. They terminate an node as soon as it can be guaranteed that no solution to the problem can be obtained by considering choices that correspond to the node's descendants.

Differences

Backtracking

[1] It is used to find all possible solutions available to the problem.
[2] It traverse tree by DFS(Depth First Search).
[3] It realizes that it has made a bad choice and undoes the last choice by backing up.
[4] It searches the state space tree until it finds a solution.
[5] It involves feasibility function.

Branch-and-Bound

[1] It is used to solve optimization problem.
[2] It may traverse the tree in any manner, DFS or BFS.
[3] It realizes that it already has a better optimal solution that the pre-solution leads to so it abandons that pre-solution.
[4] It completely searches the state space tree to get optimal solution.
[5] It involves bounding function.

Text Book

Introduction to the Design & Analysis of Algorithms, Anany Levitin, Second Edition, Pearson Education, 2007