1.a. Explain the quicksort with example.

A. Quicksort:

This technique was invented by Hoare and he is considered that his method to be a fast method to sort the elements. Here the division into two subarrays is made. so that the sorted subarrays do not need to be much. Later, this is accomplished by rearranging the elements in an array. In this method, the list is divided into two based on the pivot element. Usually, the first element is considered as the pivot element. Now move the pivot element into the correct position in the list. The elements to the left of pivot are less than the pivot and the elements to right are greater than pivot.

$i<j$: swap i^{th} element and j^{th} element

$i>j$: swap pivot element and j^{th} element

Example:

1	2	3	4	5	6	7	8	9
65	95	70	80	98	58	63	40	25
p	i							j

		3					8	
65	25	70	80	98	58	63	40	95
		i					i	

$i<j$

			4			7		
65	25	40	80	98	58	63	70	95
			i			j		

$i<j$

				5	6			
65	25	40	63	98	58	80	70	95
				i	j			

$i<j$

				5	6			
65	25	40	63	58	98	80	70	95
				j	i			

$i>j$ swap i with pivot element

58	25	40	63	65	98	80	70	95

In this example we can observe that the elements to the left of pivot element are less than pivot and the elements to the right of pivot are greater than pivot.

Algorithm:

```
Algorithm(left,right)
{
p=left, q=right
if(p<q) then
{
j=partition(a,p,q);
Quicksort(p,j-1);
Quicksort(j+1,q);
}
}
Algorithm partition(a,left,right)
{
p=a[left]; i:=left; j:=right;
repeat
{
repeat;
i:=i+1;
until(a[i]≥p)
repeat
j:=j-1;
until(a[i]≤p)
if(i<j)then
interchange(a,i,j)
}
until(i≥j)
a[left]:=a[j]
a[j]:=p;
return j;
}
Algorithm Interchange(a,i,j)
{
t:=a[i];
a[i]:=a[j];
a[j]:=t;}
repeat
i:=i+1;
until(a[i]≥p)
```

```
repeat
i = j-1;
until(a[j] ≤ p)
```

1-6) Calculate the analysis of merge sort.

Analysis:

In merge sort algorithm two recursive calls are made. each recursive call focuses on n/2 elements of the list and one call is made to combine two sublists i.e to merge all 'n' elements hence the reccurance relation is

$$c(n) = c(n/2) + c(n/2) + cn, \text{ for } n > 1$$

$$c(n) = 2c(n/2) + cn$$

$$c(1) = 0, \text{ for } n = 0$$

By using Masters's theorm

$$T(n) = a \cdot T(n/b) + F(n)$$

$a = 2, n = n, b = 2, F(n) = c \cdot n = n$

↓ c is constant

$$n^{\log_b^a} = n^{\log_2^2} = n$$

$$F(n) = n$$

$$n = n$$

$$F(n) = n^{\log_b^a}$$

Here case(ii) is satisfied

$$\text{if } F(n) = \Theta(n^{\log_b^a})$$

$$n = \Theta(n^{\log_2^2})$$

$$n = \Theta(n)$$

$$n = n$$

$$T(n) = \Theta(n^{\log_b^a} \cdot \log n)$$

$$T(n) = \Theta(n \log n)$$

Time complexity of merge sort

Substitution Method:

$$c(n) = c(n/2) + c(n/2) + c \cdot n \rightarrow ①$$

$$c(1) = 0 \rightarrow ②$$

$$c(n) = 2c(n/2) + cn \rightarrow ③$$

apply backward substitution method

$n = 2^k$ in eq ③

$c(2^k) = 2c(2^k/2) + c \cdot 2^k$

$c(2^k) = 2c(2^{k-1}) + c \cdot 2^k \longrightarrow ④$

put $k = k-1$

$c(2^{k-1}) = 2c(2^{k-2}) + c \cdot 2^{k-1} \longrightarrow ⑤$

sub eq ⑤ in ④

$c(2^k) = 2(2c(2^{k-2}) + c \cdot 2^{k-1}) + c \cdot 2^k$

$= 4c\,2^{k-2} + 2c \cdot 2^{k-1} + c \cdot 2^k$

$= c(2^k) + c \cdot 2^k + c \cdot 2^k$

$= 2^3 c(2^{k-3}) + 3c\,2^k$

$= 2^4 c(2^{k-4}) + 4c\,2^k$

⋮

$c(2^k) = 2^k \cdot c(2^{k-k}) + k \cdot c\,2^k$

$= 2^k \cdot c(1) + k \cdot c \cdot 2^k$

$= 0 + k \cdot c \cdot 2^k$

$c(2^k) = \underbrace{k c 2^k}_{\text{constants}}$

$c(2^k) = 2^k = n = n \log_n^n$

Time complexity for merge sort.

Explain the knapsack problem with example

In this method n objects are given and a knapsack (or) empty bag are given object 'i' as a weight (wi) and the knapsack as the capacity 'm'. Now the knapsack problem is that we should place the objects into the bag without exceeding bag capacity.

→ If a fraction x_i, $0 \le x_i \le 1$ of object 'i' is placed into the knapsack then a profit of $p_i x_i$ is earned.

→ The problem can be stated as maximize

$\sum_{1 \le i \le n} p_i x_i$ such that $\sum_{1 \le i \le n} w_i x_i$

Example:

Consider the following instance of knapsack problem

$n = 3, m = 20$

$(p_1, p_2, p_3) = (25, 24, 15)$

$(w_1, w_2, w_3) = (18, 15, 10)$

To solve knapsack problem there are 4 cases

case i: maximum profit

case ii: minimum weight

case iii: maximum profit per unit weight

case iv: maximum weight

case i: Maximum profit

$x_1 = 1$

$x_2 = 2/15$

$x_3 = 0$

$\sum p_i x_i = p_1 x_1 + p_2 x_2 + p_3 x_3$

$= (25)(1) + (24)\left(\frac{2}{15}\right) + (15)(0)$

$= 25 + \frac{16}{5} + 0$

$= \frac{141}{5} = 28.2$

case ii: Minimum weight

$w_3 = 10, x_3 = 1$ (completely stored)

$w_2 = 15$

$x_2 = 10/15$

$w_1 = 18$

$x_1 = 0$

$\sum w_i x_i = w_1 x_1 + w_2 x_2 + w_3 x_3$

$= (18)(0) + (15)\left(\frac{10}{15}\right) + (10)(1)$

$= 0 + 10 + 10$

$\sum w_i x_i = 20$

case iii:

Maximum profit per unit weight

$\frac{p_1}{w_1} = \frac{25}{18} = 1.4$ $\frac{p_2}{w_2} = \frac{24}{15} = 1.6$ $\frac{p_3}{w_3} = \frac{15}{10} = 1.5$

$x_1 = 1.4$ $x_2 = 1.6$ $x_3 = 1.5$

$x_1 = 0$ $x_2 = 1$ $x_3 = 1/2$

$\sum p_i x_i = p_1 x_1 + p_2 x_2 + p_3 x_3$

$= (25)(0) + (24)(1) + 15(\frac{1}{2})$

$= 24 + \frac{15}{2}$

$\sum p_i x_i = 31.5$

case iv:

Maximum weight

$x_1 = 1$

$x_2 = 2/15$

$x_3 = 0$

$\sum w_i x_i = w_1 x_1 + w_2 x_2 + w_3 x_3$

$= (18)(1) + (15)(\frac{2}{15}) + (10)(0)$

$= 18 + 2$

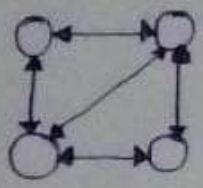
$\sum w_i x_i = 20$

1-b Solve the travelling sales person problem

To calculate travelling sales person problem the eq's

1. $g(i, \phi) = C_{i1}, 1 \leq i \leq n$
2. $g(i, S) = \min\{ C_{ij} + g(j, S - \{j\}) \}$ $j \in S$

Ex:



The cost adjacency matrix

$$\begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

$g(i,\phi) = c_{i1}\ (1 \le i \le n)$

$g(1,\phi) = c_{11} = 0$

$g(2,\phi) = c_{21} = 5$

$g(3,\phi) = c_{31} = 6$

$g(4,\phi) = c_{41} = 8$

$g(i,j) = \min\{c_{ij} + g(j, s-\{j\})\}$

$s = 1$

$g(2,3) = \min\{c_{23} + g(3,\phi)\} = \{9+6\} = 15$

$g(2,4) = \min\{c_{24} + g(4,\phi)\} = \{10+8\} = 18$

$g(3,2) = \min\{c_{32} + g(2,\phi)\} = \{13+5\} = 18$

$g(3,4) = \min\{c_{34} + g(4,\phi)\} = \{12+8\} = 20$

$g(4,2) = \min\{c_{42} + g(2,\phi)\} = \{8+5\} = 13$

$g(4,3) = \min\{c_{43} + g(3,\phi)\} = \{9+6\} = 15$

$$\begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 10 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

$s \Rightarrow 2$

$g(2,\{3,4\}) = \min\{(c_{23} + g\{3,4\}) \in c_{24} + g(4,3)\}$

$= \min(\{9+20, 10+15\} = \min\{29, 25\}) = 25$

$g(3,\{2,4\}) = \min\{(c_{32} + g\{2,4\}, (c_{34} + g(4,2)\}$

$= \min\{13+18, 12+13\}$

$= \min\{31, 25\} = 25$

$g(4,\{(2,3)\}) = \min\{(c_{42} + g\{2,3\}, (c_{43} + g(2,3)\}$

$= \min\{8+15, 9+18\}$

$= \min\{23, 27\} = 23$

$s \Rightarrow 3$

$g(1,\{1,2,3,4\}) = \min\{(c_{12} + g(2,\{3,4\})\} = (c_{13} + g\{2,4\})$

$+ g(4,\{(2,3\}\}$

$= \min\{10+25, 15+25, 20+23\}$

$= \min\{35, 40, 43\} = 35$

$1 \to 2 \to 4 \to 3 \to 1 = 35$