**Slip 8**

Q.1) Write a program to implement Fractional Knapsack problems using Greedy Method

#include <stdio.h>

void main()

{

    int capacity, no\_items, cur\_weight, item;

    int used[10];

    float total\_profit;

    int i;

    int weight[10];

    int value[10];

    printf("Enter the capacity of knapsack:\n");

    scanf("%d", &capacity);

    printf("Enter the number of items:\n");

    scanf("%d", &no\_items);

    printf("Enter the weight and value of %d item:\n", no\_items);

    for (i = 0; i < no\_items; i++)

    {

        printf("Weight[%d]:\t", i);

        scanf("%d", &weight[i]);

        printf("Value[%d]:\t", i);

        scanf("%d", &value[i]);

    }

    for (i = 0; i < no\_items; ++i)

        used[i] = 0;

    cur\_weight = capacity;

    while (cur\_weight > 0)

    {

        item = -1;

        for (i = 0; i < no\_items; ++i)

            if ((used[i] == 0) &&

                ((item == -1) || ((float) value[i] / weight[i] > (float) value[item] / weight[item])))

                item = i;

        used[item] = 1;

        cur\_weight -= weight[item];

        total\_profit += value[item];

        if (cur\_weight >= 0)

            printf("Added object %d (%d Rs., %dKg) completely in the bag. Space left: %d.\n", item + 1, value[item], weight[item], cur\_weight);

        else

        {

            int item\_percent = (int) ((1 + (float) cur\_weight / weight[item]) \* 100);

            printf("Added %d%% (%d Rs., %dKg) of object %d in the bag.\n", item\_percent, value[item], weight[item], item + 1);

            total\_profit -= value[item];

            total\_profit += (1 + (float)cur\_weight / weight[item]) \* value[item];

        }

    }

    printf("Filled the bag with objects worth %.2f Rs.\n", total\_profit);

}

Q.2) Write Program to implement Traveling Salesman Problem using nearest neighbor algorithm

/\*

\* C++ Program to Implement Traveling Salesman Problem using Nearest neighbour Algorithm

 \*/

#include<stdio.h>

#include<conio.h>

#include<iostream>

using namespace std;

int c = 0,cost = 999;

int graph[4][4] = { {0, 10, 15, 20},

                    {10, 0, 35, 25},

                    {15, 35, 0, 30},

                    {20, 25, 30, 0}

                  };

void swap (int \*x, int \*y)

{

    int temp;

    temp = \*x;

    \*x = \*y;

    \*y = temp;

}

void copy\_array(int \*a, int n)

{

    int i, sum = 0;

    for(i = 0; i <= n; i++)

    {

        sum += graph[a[i % 4]][a[(i + 1) % 4]];

    }

    if (cost > sum)

    {

        cost = sum;

    }

}

void permute(int \*a, int i, int n)

{

   int j, k;

   if (i == n)

   {

        copy\_array(a, n);

   }

   else

   {

        for (j = i; j <= n; j++)

        {

            swap((a + i), (a + j));

            permute(a, i + 1, n);

            swap((a + i), (a + j));

        }

    }

}

int main()

{

   int i, j;

   int a[] = {0, 1, 2, 3};

   permute(a, 0, 3);

   cout<<"minimum cost:"<<cost<<endl;

   getch();

}