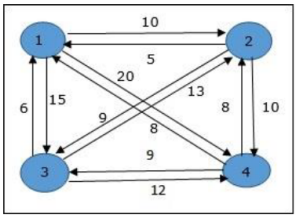
15-03-2021 Algos Lab 7 **106119100** Rajneesh Pandey

Question 1 -

1. In the traveling salesman Problem, a salesman must visit n cities. We can say that salesman wishes to make a tour or Hamiltonian cycle, visiting each city exactly once and finishing at the city he starts from. There is a non-negative cost c (i, j) to travel from the city i to city j. The goal is to find a tour of minimum cost. We assume that every two cities are connected. We can model the cities as a complete graph of n vertices, where each vertex represents a city.

106119100

Write a Program to implement Travelling Salesman Problem for the following example.




```cpp
//106119100 Rajneesh Pandey

#include <bits/stdc++.h>
using namespace std;
void display_path(vector<int> path){
    cout << "DISPLAYING PATH" << endl;
    for (int a : path)
       cout << a << " ";
    cout << endl;
}
int helper(vector<vector<int>> &grid, map<int, bool> &m, int st){
    int cost = INT_MAX;
    bool flag = false;
    m[st] = true;
    for (int i = 0; i < grid.size(); i++){
        int mini = -1;
        if (!m[i] && i != st)
        {
            mini = grid[st][i];
            mini += helper(grid, m, i);
            flag = true;
            if (mini < cost)
                cost = mini;
        }
    }
    m[st] = false;
    if (flag)
        return cost;
    return grid[st][0];
}
int main()
{
    vector<vector<int>> grid = {
          {0, 10, 15, 20},
          {5, 0, 9, 10},
          {6, 13, 0, 12},
          {8, 8, 9, 0}};
    map<int, bool> m;
    int cost = helper(grid, m, 0);
    cout << cost;
}
```

Time(sec) : 0 Memory(MB) : 3.3262545483398

Output: Copy

```
35
```

Question 2

2. Write a program to implement Yet Another String Matching Problem

Suppose you have two strings s and t, and their length is equal. You may perform the following operation any number of times: choose two different characters c1 and c2, and replace every occurrence of c1 in both strings with c2. Let's denote the distance between strings s and t as the minimum number of operations required to make these strings equal. For example, if s is abcd and t is ddcb, the distance between them is 2 — we may replace every occurrence of a with b, so s becomes bbcd, and then we may replace every occurrence of b with d, so both strings become ddcd. You are given two strings S and T. For every substring of S consisting of |T| characters you have to determine the distance between this substring and T.

```cpp
//106119100 Rajneesh Pandey

#include <bits/stdc++.h>

using namespace std;
int f[6], ans, sl, tl, a[6][6];
char s[125005], t[125005];
int find(int x)
{
    return x == f[x] ? x : f[x] = find(f[x]);
}
int main()
{
    cin >> s >> t;
    sl = strlen(s);
    tl = strlen(t);
    for (int i = 0; i <= sl - tl; i++)
    {
        ans = 0;
        for (int j = 0; j < 6; j++)
            f[j] = j;
        for (int j = 0; j < tl; j++)
            a[s[i + j] - 'a'][t[j] - 'a'] = 1;
        for (int j = 0; j < 6; j++)
            for (int k = 0; k < 6; k++)
                if (a[j][k])
                {
                    if (find(j) != find(k))
                        f[find(j)] = find(k), ans++;
                    a[j][k] = 0;
                }
        cout << ans;
    }
    return 0;
}
```