

-----

**NAME: Rajneesh Pandey**

**ROLL NO: 106119100**

## **Algos Lab - 5**

*Q1: Given a weighted undirected graph. Write a program to Find the sum of weights of edges of a Minimum Spanning Tree using Prims Algorithm.*

*Q2: Write a program to implement knapsack problem using greedy method*

*1. Given a set of items, each with a weight and a value.*

*2. Determine the number of each item to include in a collection so that the total weight is less than a given limit and the total value is as large as possible.*

*3. It derives its name from the problem faced by someone who is constrained by a fixed-size knapsack and must fill it with the most useful items.*

## Question 1

```
#include <iostream>
#include <vector>
#include <queue>
#include <map>
#include <unordered_map>

using namespace std;

int prims(int n, int e)
{
    vector<int> vec;
    map<int, vector<pair<int, int>>> g;
    unordered_map<int, bool> vis;
    for (int i = 0; i < e; i++)
    {
        int a, b, w;
        cin >> a;
        cin >> b;
        cin >> w;
        pair<int, int> p1 = {a, w};
        pair<int, int> p2 = {b, w};
        g[a].push_back(p2);
        g[b].push_back(p1);
    }

    int count = 1;
    int result = 0;
    vec.push_back(g.begin()->first);
    vis[g.begin()->first] = true;

    while (count != n)
    {
        int min = INT32_MAX;
        int node;
        for (auto ele : vec)
        {
            vector<pair<int, int>> nb = g[ele];
            for (auto a : nb)
            {
                if (min >= a.second && !vis[a.first])
                {
                    min = a.second;
                    node = a.first;
                }
            }
        }
    }
}
```

```
        }  
    }  
    }  
    vec.push_back(node);  
    vis[node] = true;  
    count++;  
    result += min;  
    }  
    return result;  
}  
int main()  
{  
    int node, edge;  
    cin >> node;  
    cin >> edge;  
    cout << prims(node, edge);  
}
```

## Input

```
4 5  
1 2 7  
1 4 6  
4 2 9  
4 3 8  
2 3 6
```

## Output

```
19
```

## Question 2

```
#include <iostream>
#include <queue>
#include <vector>
#include <iomanip>

using namespace std;

void printVec(vector<double> vec){
    for(double &ch : vec){
        cout << fixed << setprecision(6) << ch << " ";
    }
    cout << endl;
    return;
}

double knapsack(vector<pair<double, double>> vec, int cap){
    priority_queue<pair<double, pair<double, double>>> pq;
    double sum = 0.0;
    double profit = 0.0;
    for(const auto &ch : vec){
        pq.push({ch.second/ch.first, ch});
    }
    vector<double> result(vec.size(), 0);
    int count = 0;
    while(sum != cap && !pq.empty()){
        pair<double, pair<double, double>> ch = pq.top();
        if(sum + ch.second.first < cap){
            sum += ch.second.first;
            profit += ch.second.second;
            result[count] = 1;
            count++;
        }
        else if(sum + ch.second.first > cap){
            profit += (ch.second.second*((cap-sum)/ch.second.first));
            result[count] = (cap-sum)/ch.second.first;
            sum += (cap-sum);
            count++;
        }

        pq.pop();
    }
    printVec(result);
    return profit;
}
```

```

}

int main(){
    int n;
    cout << "Enter number of objects"<<endl;
    cin >> n;
    int wt,pt;
    vector <pair<double, double>> vec;
    cout << "Enter the wts and profits of each object"<<endl;
    while(n--){
        cin >> wt >> pt;
        vec.push_back({wt,pt});
    }
    cout << "Enter the capacity of knapsack"<<endl;
    int cap;
    cin >> cap;

    cout << knapsack(vec, cap);
    return 0;
}

```

## Input

```

Enter number of objects
7
Enter the wts and profits of each object
2 10
3 5
5 15
7 7
1 6
4 18
1 3
Enter the capacity of knapsack
15

```

## Output

```

1.000000 1.000000 1.000000 1.000000 1.000000 0.666667 0.000000
55.333333

```