Прізвище: Долінський
Ім'я: Олег
Група: КН-406
Варіант: 8
Кафедра: САПР
Дисципліна: Дискретні моделі в САПР
Прийняв: Кривий Р.З.
Посилання: https://github.com/olehdol/labs.git

ЗВІТ
до лабораторної роботи №1
на тему «**Побудова мінімального остового дерева**»

Мета роботи

Отримати практичні навички роботи з пошуку мінімального остового дерева.

Короткі теоретичні відомості

Надамо множині E значення величини $\emptyset$ — порожньої множини.

Визначимо, які з ребер початкового графа, що не належать до E, при долученні до E не утворюють циклів.

Якщо такі ребра є, то:

- серед цих ребер визначимо «найлегше», тобто з найменшою вагою;
- долучимо це ребро до множини E;
- переходимо до виконання пункту 2.

Якщо таких ребер немає, то припиняємо побудову мінімального остовного дерева.

Індивідуальне завдання

```
8
0 0 7 0 0 0 46 98
0 0 33 0 0 99 0 0
7 33 0 99 92 28 0 64
0 0 99 0 15 52 0 0
0 0 92 15 0 0 0 58
0 99 28 52 0 0 0 0
46 0 0 0 0 0 0 36
98 0 64 0 58 0 36 0
```

Виконання

Задаю модель графу, модель ребра та модель підмножини.

Далі потрібно перетворити вхідну матрицю у відмовідні моделі:

```
Мінімальне кістякове дерево Прима
Початкова матриця:
        -1      0       7       0       0       0       46      98
        0       -1      33      0       0       99      0       0
        7       33      -1      99      92      28      0       64
        0       0       99      -1      15      52      0       0
        0       0       92      15      -1      0       0       58
        0       99      28      52      0       -1      0       0
        46      0       0       0       0       0       -1      36
        98      0       64      0       58      0       36      -1
```

```
                        Початковий граф:
Кількість ребер: 13
Кількість вершин: 8

Джерело Ціль    Вага
0       2       7
3       4       15
2       5       28
1       2       33
6       7       36
0       6       46
3       5       52
4       7       58
2       7       64
2       4       92
0       7       98
1       5       99
2       3       99
```

Сортую ребра відносно їхньої ваги.

```
Array.Sort(graph.Edges, (Edge first, Edge second) => {
    return first.Weight.CompareTo(second.Weight);
});
```

Визначення прилежності до різних наборів:

```
private static int Find(Subset[] subsets, int i)
{
    if (subsets[i].Parent != i)
        subsets[i].Parent = Find(subsets, subsets[i].Parent);
    return subsets[i].Parent;
}
```

Об’єднання вершин.

```
int xroot = Find(subsets, x);
int yroot = Find(subsets, y);

if (subsets[xroot].Rank < subsets[yroot].Rank)
    subsets[xroot].Parent = yroot;
else if (subsets[xroot].Rank > subsets[yroot].Rank)
    subsets[yroot].Parent = xroot;
else
{
    subsets[yroot].Parent = xroot;
    subsets[xroot].Rank++;
}
```

Та основний алгоритм проходу по всім вершинам графу:

```
int verticesCount = graph.VerticesCount;
Edge[] edgesArray = new Edge[verticesCount];

Array.Sort(graph.Edges, (Edge first, Edge second) => {
    return first.Weight.CompareTo(second.Weight);
});

Subset[] subsets = new Subset[verticesCount];

for (int v = 0; v < verticesCount; v++)
    subsets[v] = new Subset() { Parent = v, Rank = 0 };

int e = default;
int i = default;
while (e < verticesCount - 1)
{
    Edge nextEdge = graph.Edges[i++];
    int x = Find(subsets, nextEdge.Source);
    int y = Find(subsets, nextEdge.Destination);

    if (x != y)
    {
        if (nextEdge != null)
            edgesArray[e++] = nextEdge;
        Union(subsets, x, y);
    }
}
```

Результат:

```
                Новий мінімізований граф:
Кількість ребер: 7
Кількість вершин: 8

Джерело Ціль    Вага
0       2       7
3       4       15
2       5       28
1       2       33
6       7       36
0       6       46
3       5       52
```

Висновок:

Розглянуто алгоритм пошуку та побудови мінімального кістякового дерева зваженого неорієнтованого графа за допомогою алгоритму Прима. Алгоритм Прима починається з побудови виродженого лісу, що містить V дерев, кожне з яких складається з однієї вершини. Далі виконуються операції об'єднання двох дерев, для чого використовуються найлегші можливі ребра, поки не утвориться єдине дерево.