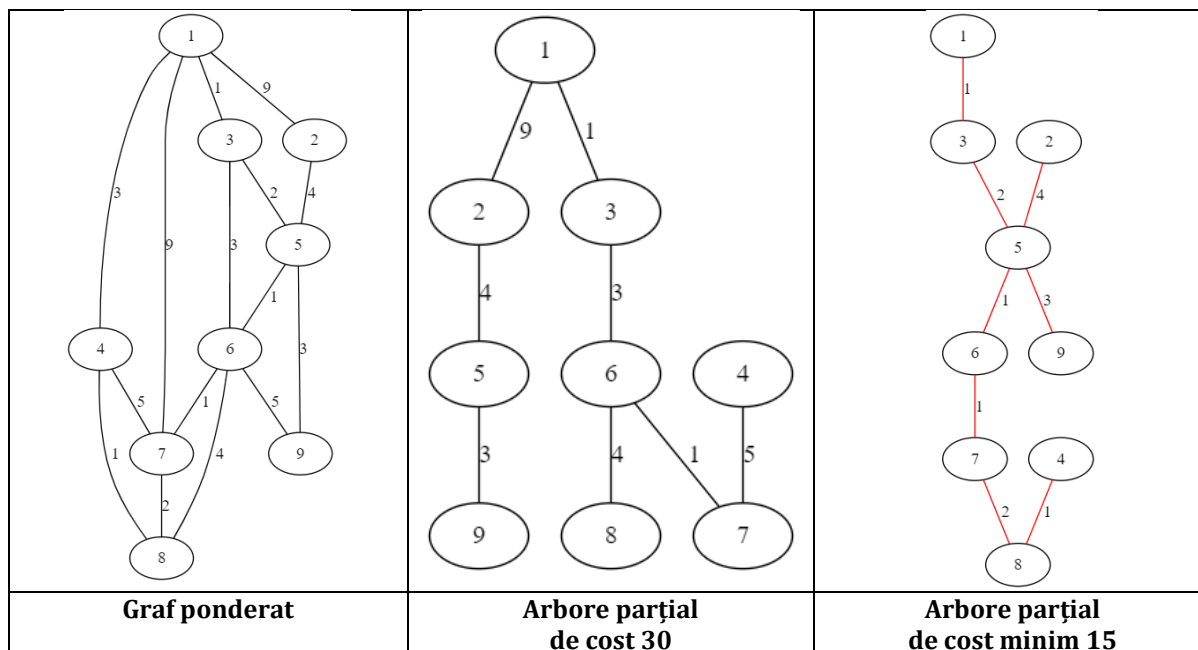


TEHNICA DE PROGRAMARE "GREEDY" (continuare)

1. Determinarea unui arbore parțial de cost minim (algoritmul lui Kruskal)

Considerăm un graf neorientat ponderat conex G (i.e., muchiile au asociate costuri). Să se determine un arbore parțial de cost minim.



Un *arbore* este un graf conex și aciclic.

Teoremă: Orice arbore cu $n \geq 1$ vârfuri are $n - 1$ muchii.

Un *arbore parțial* este un arbore care conține toate cele n vârfuri ale unui graf, deci are $n - 1$ muchii. Altfel zis, un arbore parțial al unui graf este un graf care are aceleași vârfuri și un număr minim de muchii astfel încât să rămână conex, precum și un număr maxim de muchii astfel încât să fie aciclic. Practic, un arbore parțial de cost minim reprezintă un graf conex care se obține eliminând un număr maxim de muchii din graful inițial, este aciclic și, în plus, costul său este minim (i.e., orice alt arbore parțial al grafului respectiv are un cost cel puțin egal cu al său).

Costul unui *arbore parțial* este suma costurilor muchiilor sale.

Algoritmul lui Kruskal

1. Se sortează muchiile grafului inițial crescător după costuri:

$$G = (V, E)$$

$$V = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$E = \{(1,3,1), (6,7,1), (4,8,1), (5,6,1), (3,5,2), (7,8,2), (5,9,3), (1,4,3), (3,6,3), (6,8,4), (2,5,4), (4,7,5), (6,9,5), (1,2,9), (1,7,9)\}$$

2. Se consideră graful parțial format din cele $n=9$ vârfuri ale grafului G și nicio muchie (i.e., fiecare vârf este izolat și formează o componentă conexă):

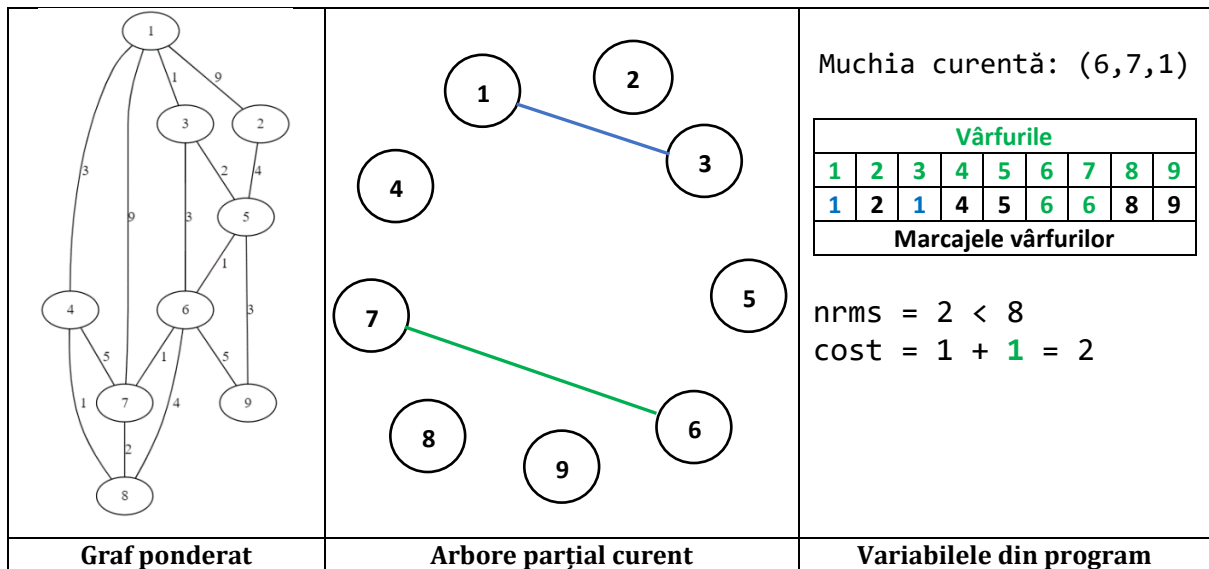
		<p>L = tablou unidimensional cu n=9 elemente de tip int</p> <p>L[i] = marcajul vârfului i = numărul componentei conexe din care face parte vârful i = un vârf din respectiva componentă conexă</p> <p>nrms = numărul muchiilor selectate</p> <p>cost = costul total APM</p> <table border="1"><tr><th colspan="9">Vârfurile</th></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr><tr><th colspan="9">Marcajele vârfurilor</th></tr></table> <p>nrms = 0</p> <p>cost = 0</p>	Vârfurile									1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	Marcajele vârfurilor								
Vârfurile																																						
1	2	3	4	5	6	7	8	9																														
1	2	3	4	5	6	7	8	9																														
Marcajele vârfurilor																																						
Graf ponderat	Arbore parțial curent	Variabilele din program																																				

3. Se parcurg muchiile grafului G și se verifică dacă muchia curentă $[x, y]$ formează ciclu cu muchiile deja selectate (i.e., $[x, y]$). În caz negativ, se selectează muchia curentă în APM. Se repetă acest pas până se vor selecta $n-1 = 8$ muchii (i.e., $nrms = 8$)

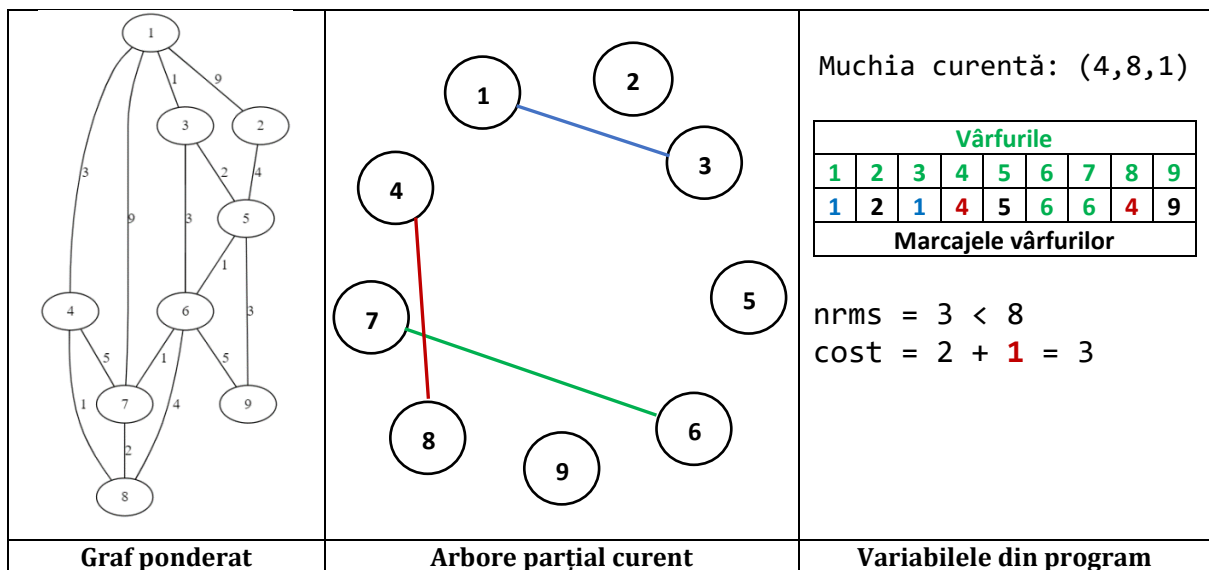
$E = \{(1,3,1), (6,7,1), (4,8,1), (5,6,1), (3,5,2), (7,8,2), (5,9,3), (1,4,3), (3,6,3), (6,8,4), (2,5,4), (4,7,5), (6,9,5), (1,2,9), (1,7,9)\}$

		<p>Muchia curentă: (1,3,1)</p> <table border="1"><tr><th colspan="9">Vârfurile</th></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr><tr><td>1</td><td>2</td><td>1</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr><tr><th colspan="9">Marcajele vârfurilor</th></tr></table> <p>nrms = 1 < 8</p> <p>cost = 0 + 1 = 1</p>	Vârfurile									1	2	3	4	5	6	7	8	9	1	2	1	4	5	6	7	8	9	Marcajele vârfurilor								
Vârfurile																																						
1	2	3	4	5	6	7	8	9																														
1	2	1	4	5	6	7	8	9																														
Marcajele vârfurilor																																						
Graf ponderat	Arbore parțial curent	Variabilele din program																																				

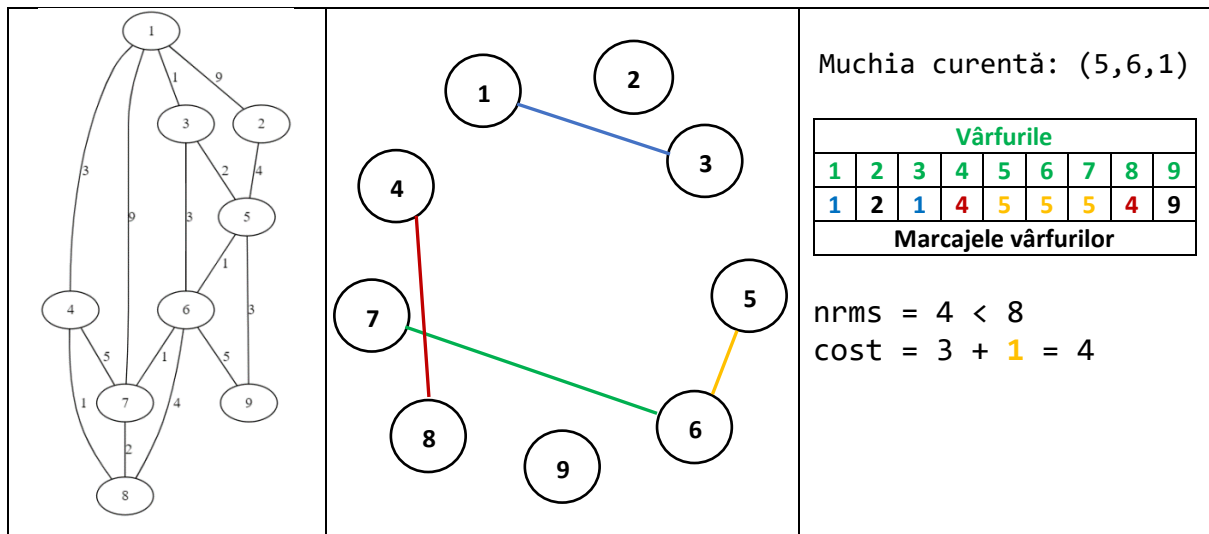
$E = \{(\underline{1,3,1}), (\underline{6,7,1}), (4,8,1), (5,6,1), (3,5,2), (7,8,2), (5,9,3), (1,4,3), (3,6,3), (6,8,4), (2,5,4), (4,7,5), (6,9,5), (1,2,9), (1,7,9)\}$



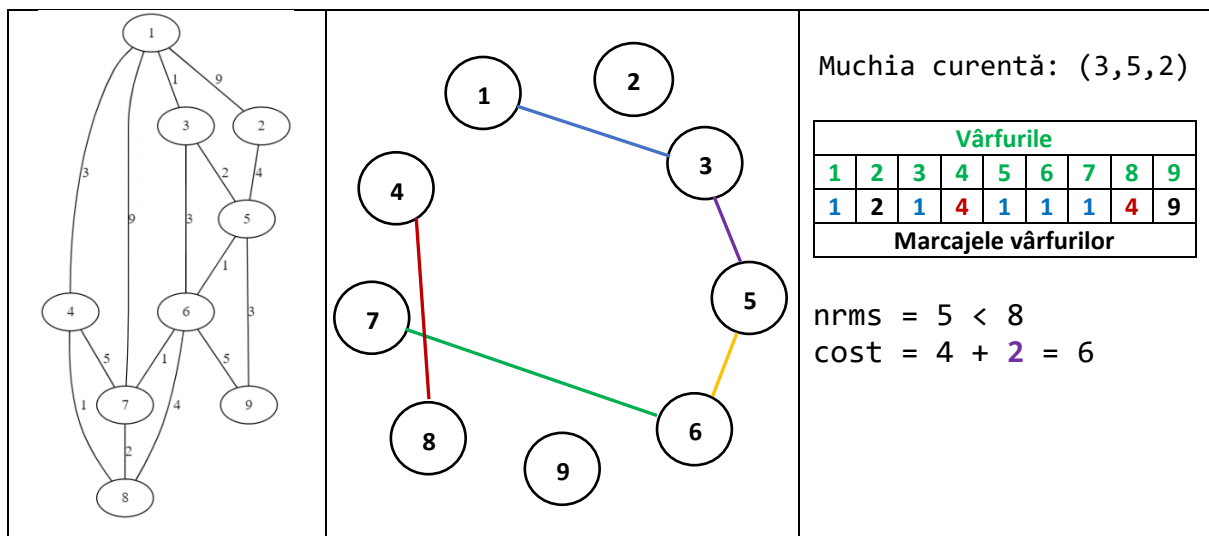
$E = \{(\underline{1,3,1}), (\underline{6,7,1}), (\underline{4,8,1}), (5,6,1), (3,5,2), (7,8,2), (5,9,3), (1,4,3), (3,6,3), (6,8,4), (2,5,4), (4,7,5), (6,9,5), (1,2,9), (1,7,9)\}$

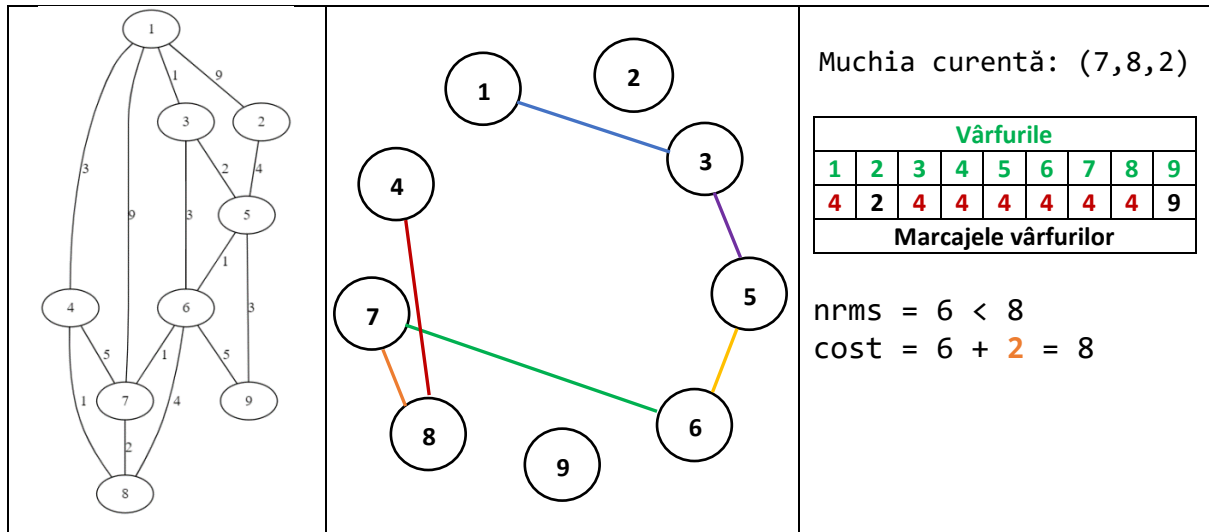
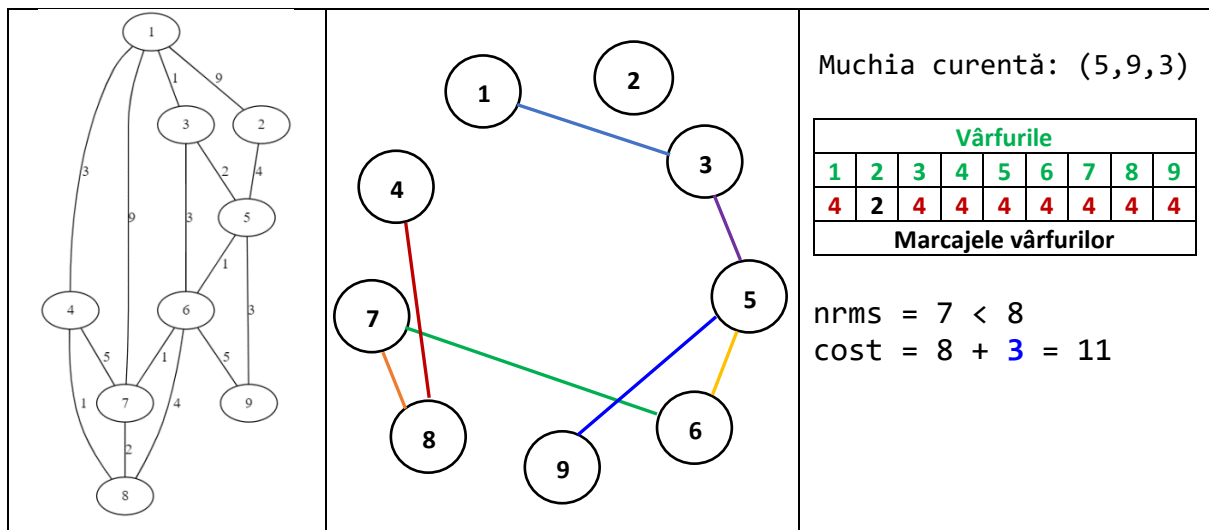


$E = \{(\underline{1,3,1}), (\underline{6,7,1}), (\underline{4,8,1}), (\underline{5,6,1}), (3,5,2), (7,8,2), (5,9,3), (1,4,3), (3,6,3), (6,8,4), (2,5,4), (4,7,5), (6,9,5), (1,2,9), (1,7,9)\}$



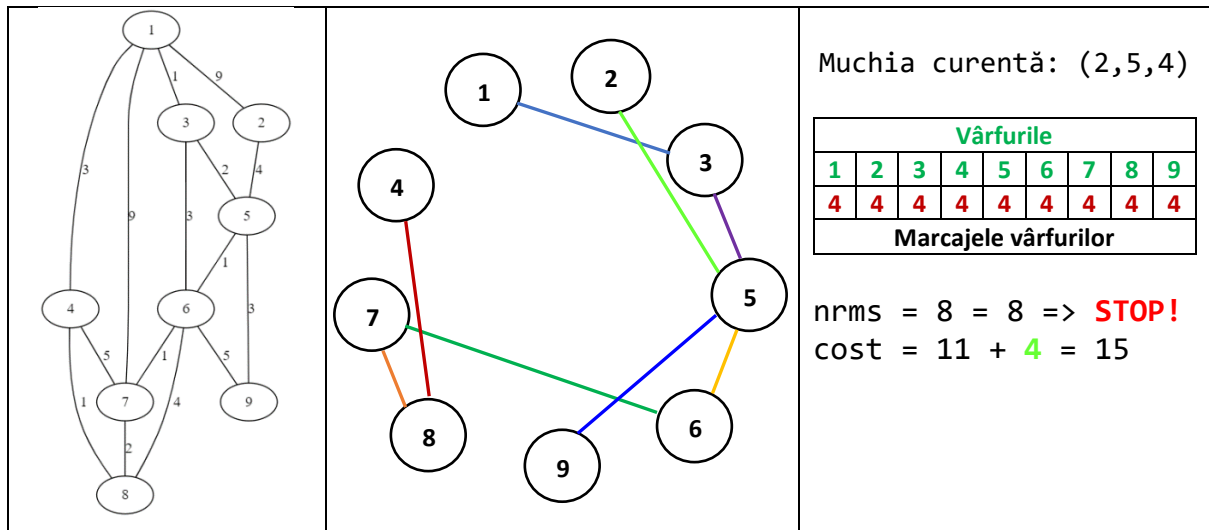
$E = \{(\underline{1,3,1}), (\underline{6,7,1}), (\underline{4,8,1}), (\underline{5,6,1}), (\underline{3,5,2}), (7,8,2), (5,9,3), (1,4,3), (3,6,3), (6,8,4), (2,5,4), (4,7,5), (6,9,5), (1,2,9), (1,7,9)\}$



$$E = \{(\underline{1,3,1}), (\underline{6,7,1}), (\underline{4,8,1}), (\underline{5,6,1}), (\underline{3,5,2}), (7,8,2), (5,9,3), (1,4,3), (3,6,3), (6,8,4), (2,5,4), (4,7,5), (6,9,5), (1,2,9), (1,7,9)\}$$

$$E = \{(\underline{1,3,1}), (\underline{6,7,1}), (\underline{4,8,1}), (\underline{5,6,1}), (\underline{3,5,2}), (\underline{7,8,2}), (5,9,3), (1,4,3), (3,6,3), (6,8,4), (2,5,4), (4,7,5), (6,9,5), (1,2,9), (1,7,9)\}$$


$E = \{ \langle \underline{1}, \underline{3}, \underline{1} \rangle, \langle \underline{6}, \underline{7}, \underline{1} \rangle, \langle \underline{4}, \underline{8}, \underline{1} \rangle, \langle \underline{5}, \underline{6}, \underline{1} \rangle, \langle \underline{3}, \underline{5}, \underline{2} \rangle, \langle \underline{7}, \underline{8}, \underline{2} \rangle, \langle \underline{5}, \underline{9}, \underline{3} \rangle, \langle \underline{1}, \underline{4}, \underline{3} \rangle, \langle \underline{3}, \underline{6}, \underline{3} \rangle, \langle \underline{6}, \underline{8}, \underline{4} \rangle, \langle \underline{2}, \underline{5}, \underline{4} \rangle, (4,7,5), (6,9,5), (1,2,9), (1,7,9) \}$

Cele 3 muchii nu pot fi selectate, deoarece
 ar forma cicluri cu muchiile deja alese!



Implementare:

graf.in	graf.out
9 15 1 2 9 1 3 1 1 7 9 1 4 3 2 5 4 3 5 2 3 6 3 4 7 5 4 8 1 5 6 1 5 9 3 6 7 1 6 9 5 6 8 4 7 8 2	Muchiile APM-ului: 1 3 1 4 8 1 5 6 1 6 7 1 3 5 2 7 8 2 5 9 3 2 5 4 Cost total APM: 15

ATENȚIE!!!

$L = (7, \underline{7}, \underline{1}, \underline{1}, \underline{1}, \underline{1}, 7, 7, \underline{7}, 7, 7)$
 Muchia curentă = [2, 9] =>
 $L[\text{muchie_curentă}.x] = 7$ și $L[\text{muchie_curentă}.y] = 7$

```

for(j = 1; j <= n; j++)
  if(L[j] == L[muchie_curentă.x])
    L[j] = L[muchie_curentă.y]
  
```

```

#include <iostream>
#include <fstream>
#include <algorithm>

using namespace std;

struct Muchie
{
    int x, y;
    float cost;
};

bool cmpMuchiiCosturi(Muchie a, Muchie b)
{
    return a.cost < b.cost;
}

int main()
{
    int n, m;
    ifstream fin("graf.in");
    fin >> n >> m;

    Muchie *tablouMuchii = new Muchie[m];

    for (int i = 0; i < m; i++)
        fin >> tablouMuchii[i].x >> tablouMuchii[i].y >>
tablouMuchii[i].cost;
    fin.close();

    sort(tablouMuchii, tablouMuchii + m, cmpMuchiiCosturi);

    int nrMuchiiSelectate = 0;
    float costTotal = 0.0;
    int *L = new int[n + 1];

    for (int i = 1; i <= n; i++)
        L[i] = i;

    ofstream fout("graf.out");
    fout << "Muchiile APM-ului: " << endl;

    for (int i = 0; i < m; i++)
        if (L[tablouMuchii[i].x] != L[tablouMuchii[i].y])
        {
            fout << tablouMuchii[i].x << " " << tablouMuchii[i].y << " "
<< tablouMuchii[i].cost << endl;
            int marcajX = L[tablouMuchii[i].x];
            int marcajY = L[tablouMuchii[i].y];
            for (int j = 1; j <= n; j++)
            {
                if (L[j] == marcajX)
                {

```

```

        L[j] = marcajY;
    }
}
nrMuchiiSelectate++;
costTotal += tablouMuchii[i].cost;
if (nrMuchiiSelectate == n - 1)
    break;
}
fout << "Cost total APM: " << costTotal << endl;
fout.close();

delete[] tablouMuchii;
delete[] L;

return 0;
}

```