**Tema Laborator 4**

**Ghimpu Lucian Eduard**

**Grupa 213**

**Documentatie**

1. **Textul problemei.**

Sa se elaboreze un algoritm care gaseste pentru un graf de forma G=(X,U), unde X este multimea varfurilor si U multimea arcelor, toate multimile interior stabile maximale.

1. **Dezvoltarea algoritmului.**
2. Citirea matricii de costuri si generarea automata in memodire a unei matrici de adiacenta dupa regula:

Ai,j=

1. Gasirea tuturor multimilor stabile interioare prin backtracking

Fie graful orientat G=(X,U). Mulţimea S ⊂ X se numeşte mulţime stabilă interior (independentă) dacă şi numai dacă ∀i∈S are loc Γi ∩S = Φ

1. Eliminarea solutiilor ca nu sunt maximale
2. Tiparirea multimilor stabile interioare maximale
3. **Descrierea algoritmului in pseudocod**

Vom descrie cele 2 etape care determina solutiile:

1. Determinarea multimilor stabile interioare prin backtracking
2. Determinarea multimilor maximale

Prin matAd[i][j] -> ne referim la matricea de adiacenta

Pentru a construii multimile stabil interioare prin backtracking vom folosi urmatoarele metode:

**Subalgortim consistent** (x,n,matAdi) este:

{x -> solutia temporara, n -> numarul de varfuri}

Daca lungime(x) <= n atunci:

Consistent=Fals

Altfel: consistent=Adevarat

{o solutie este valida atat timp cat lungime ei nu depaseste numarul de varfuri}

**Subalgoritm solutie**(x,n,matAdi) este:

{x -> solutia temporara, n -> numarul de varfuri}

Pentru i<-1,n:

Pentru j<-1,n:

Daca matAdi[x[i]][x[j]]==1 atunci:

Solutie=Fals

{verificam ca intre 2 varfuri sa nu existe muchie, astfel in cat multimea sa fie stabila}

**Subalgoritm back**(x,n,matAdi) este:

Solutii={} {array cu toate solutiile}

Cat timp @lungime(x)>0 executa:

Ales=Fals

Cat timp Ales != Adevarat si @ultimulElement(x) < n-1 executa:

x[lungimex(x)]= x[lungimex(x)]+1

ales = consistent(x,n,matAdi)

Daca ales = Adevarat atunci:

Daca solutie(x,n,matAdi) atunci:

@adauga la solutii, solutia actuala x

@adauga lui x inca o valoare

Altfel:

@x ia valoare lui x fara ultimul element (mergem inapoi sa cautam alte solutii)

Back<-solutii

In final din solutiile obtinute, vom pastra doar cele maximale.

1. **Demonstrarea corectitudinii algoritmului**

Am folosit informatii gasite in bibliografia cursului si in resursele laboratorului

1. **Date test**

Matrice adiacenta:

[0,1,0,0,0,1,0]

[0,0,0,0,0,0,1]

[0,1,0,1,0,0,0]

[0,0,0,0,0,0,1]

[0,0,0,1,0,1,0]

[0,0,0,0,0,0,1]

[1,0,1,0,1,0,0]

Solutii: {1,3,5} {2,4,6}

Matrice adiacenta:

[0,1,0]

[0,0,1]

[0,0,0]

Solutii: {1,3}