

Tema II - Ex. 1

Ichim Teodora & Radu Mihai-Emilian - 2B1

Consideram ca avem graful retinut ca lista de adiacenta si ca avem un vector c unde $c[i]$ contine varful adiacent conectat cu varful i in multumea muchiilor alese pentru cuplaj.

```
for ( $i \in |V|$ ) do vizitat[i]=0
vizitat[1]=1
coada.push=1
while (coada !=empty) do
    s=coada.front()
    afiseaza s
    if (vizitat[c[i]]!=1) do
        vizitat[c[i]]=1
        coada.push(c[i])
    for ( $i \in \text{lista}(s)$ ) do
        if (vizitat[i]!=1) do
            vizitat[i]=1
            coada.push(i)
        if (c[i]!=1) do
            vizitat[c[i]]=1
            coada.push(c[i])
coada.pop()
```

Algoritmul prezentat este o variatie a algoritmului *BFS*, deci va genera un arbore partial.

Atunci cand vizitam un nod vom vizita si nodul conectat de el printr-o muchie aleasa pentru cuplajul perfect.

Asta inseamna ca in arborele partial vom pastra muchia dintre ele deci cele doua noduri ar trebui sa fie in partiti diferite.

G are un cuplaj perfect $\implies |G|$ este par \implies ambele partiti vor avea numar egal de noduri.

Cum constructia unui arbore partial inseamna eliminarea circuitelor \implies nu vor exista muchii intre varfuri partitiilor facute anterior \implies arborele T va fi bipartit.