Drum elementar = nu se repeta noduri

Drum simplu = nu se repeta muchii

Graf partial = putem elimina muchii

Subgraf = putem elimina noduri

Graf biconex -> graf care nu are noduri critice

Graf tare conex -> graf in care exista drum de la x la y, oricare ar fi x si y varfuri

Nr max muchii graf bipartit: par(n^2 / 4) impar(n + 1)(n - 1) / 4

Graf transpus (doar pe graf orientat) = inversam sensul muchiilor

Graf complementar = daca avem muchie -> eliminam

= daca nu aveam original -> add

V = vertices;

E = edges;

Complexitate BFS and DFS -> O(V + E)

BFS -> arborele de inaltime minima

Muchie critica -> niv\_min[j] > niv[i], unde j este fiu a lui i.

Nod critic -> radacina -> are cel putin 2 fii in arborele DF

-> alt nod -> are cel putin un fiu j cu niv\_min[j] >= niv[i]

Complexitate algoritm muchii/noduri critice: O(V + E)

Complexitate Sortare Topologica -> O(V + E)

Complexitate Kruskal -> O(E log V)

Complexitate Prim -> O(V^2) or O(E log V)

Complexitate Dijkstra -> O(V^2) or O(E log V)

Complexitate Bellman-Ford -> O(E \* V)

Complexitate Distante minime in DAG -> O(V+E)

Complexitate Floyd-Warshall -> O(V^3)

Complexitate Edmonds-Karp -> O(V\*E^2)

Complexitate algoritmul lui Mickey Mouse ( Hierholzer ) -> O(E)

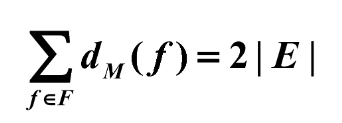
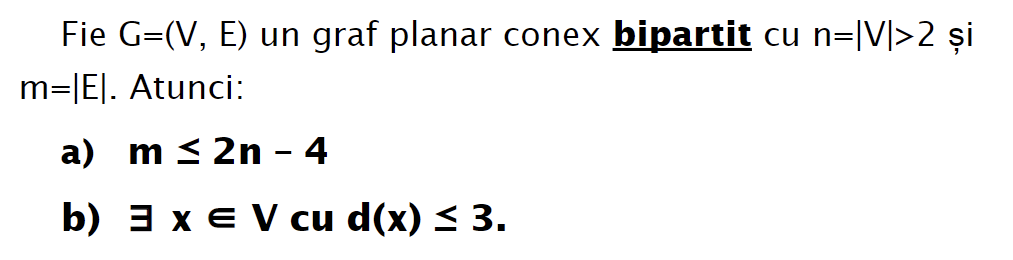
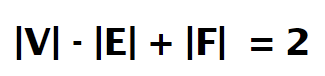
Complexitate algoritmul lui Tarjan -> O(V+E)

Taietura in retea -> este o bipartitie a multimii varfurilor a.i. Sursa si destinatia sunt in partitii diferite

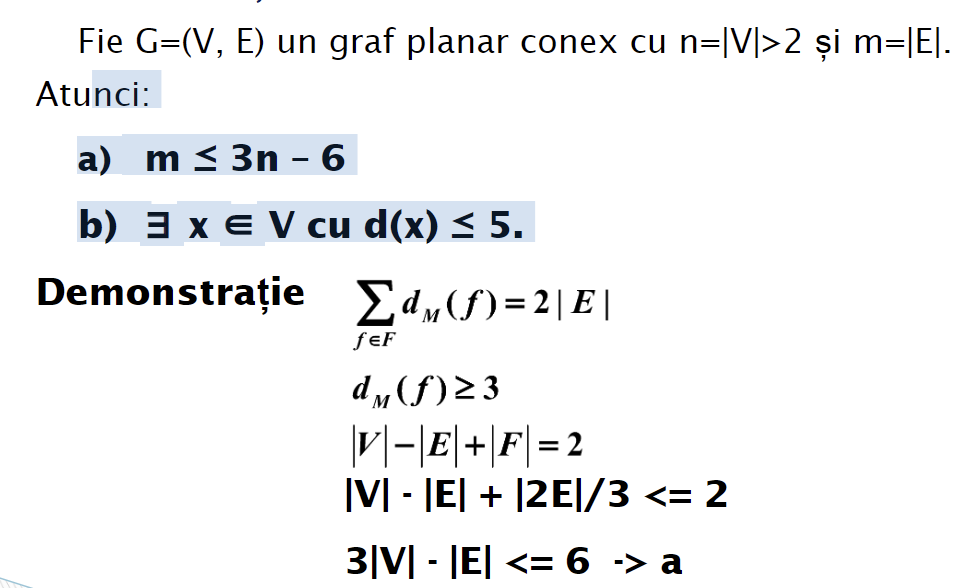
Daca graful nu este biconex => nu este hamiltonian

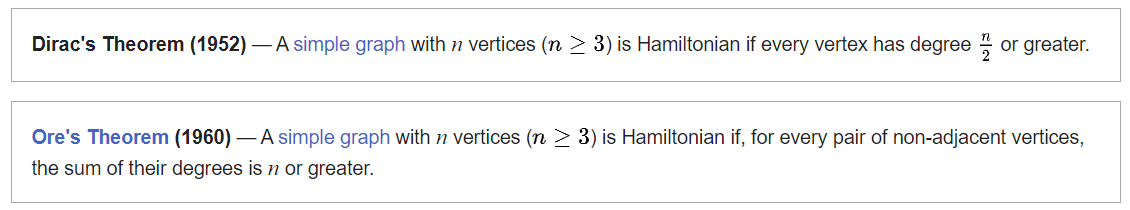
Graf eulerian -> are toate nodurile de grad par

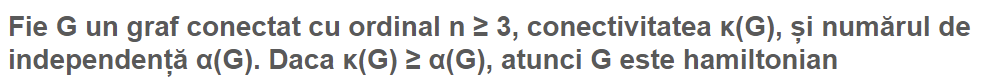
Algoritm verificare graf hamiltonian -> O(n^2 \* 2^n)

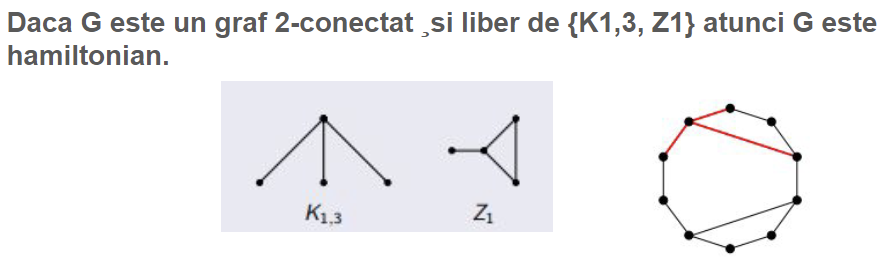
 

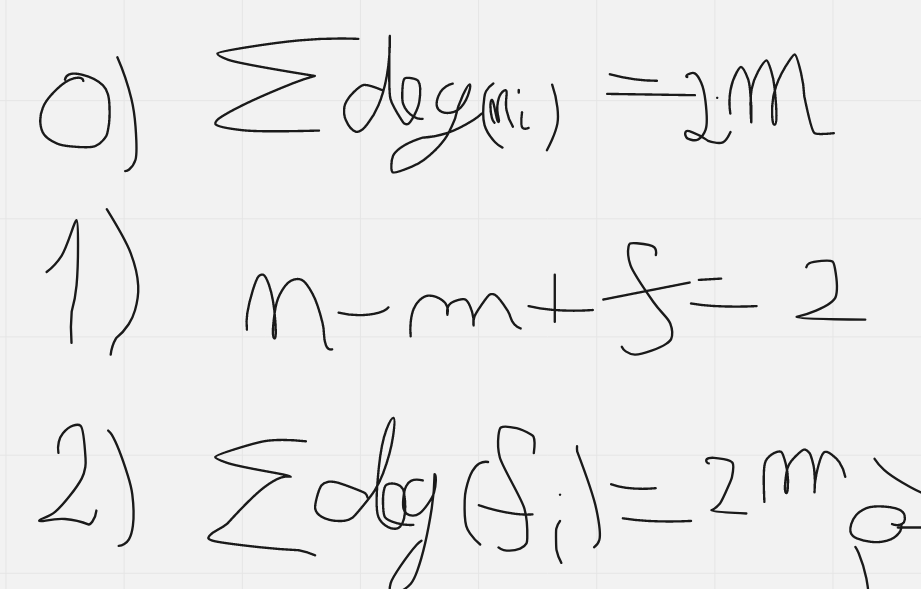


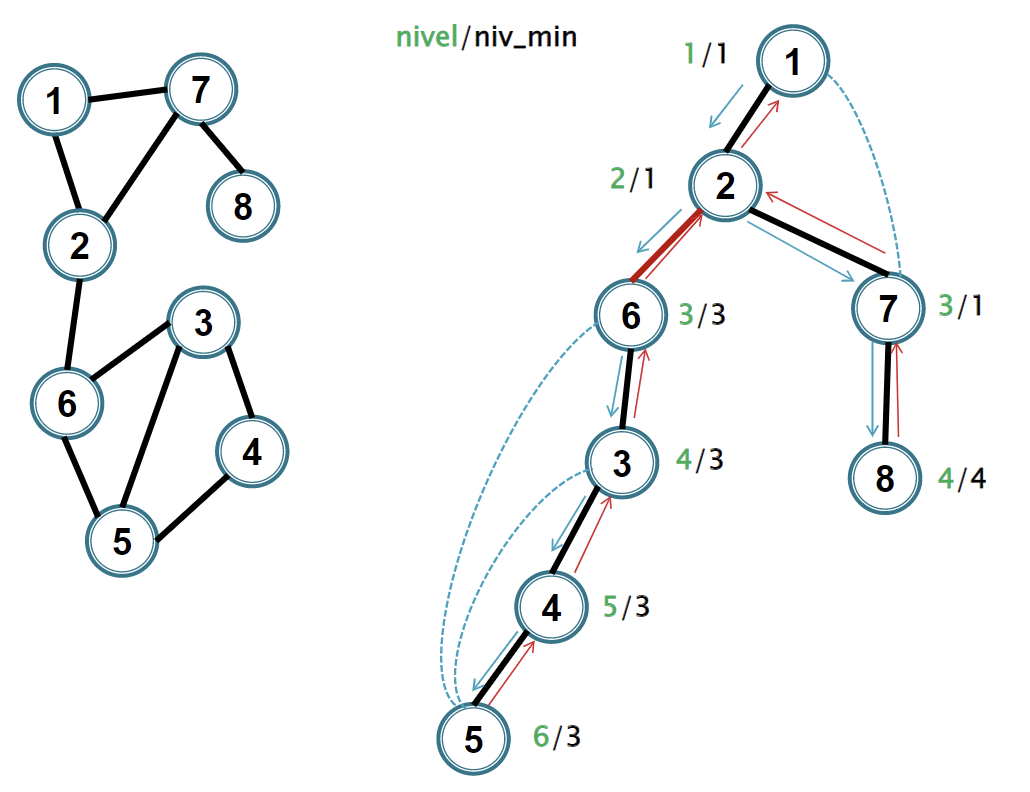


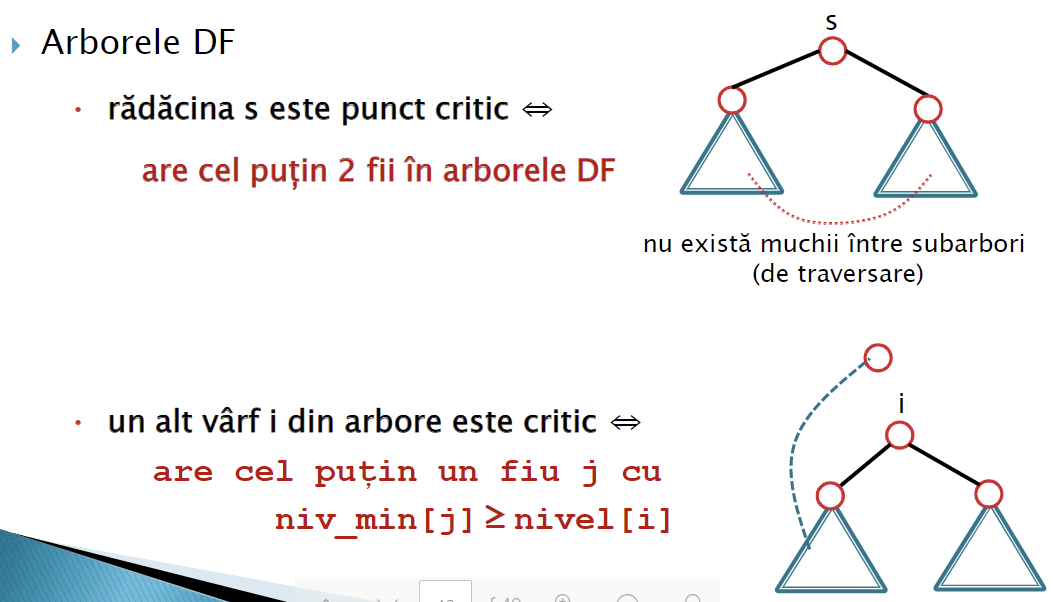


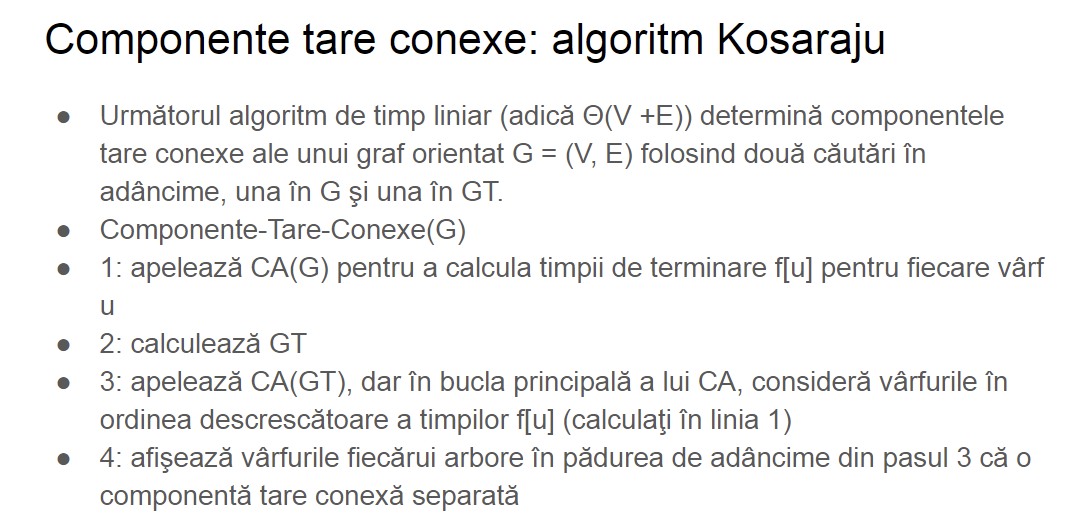


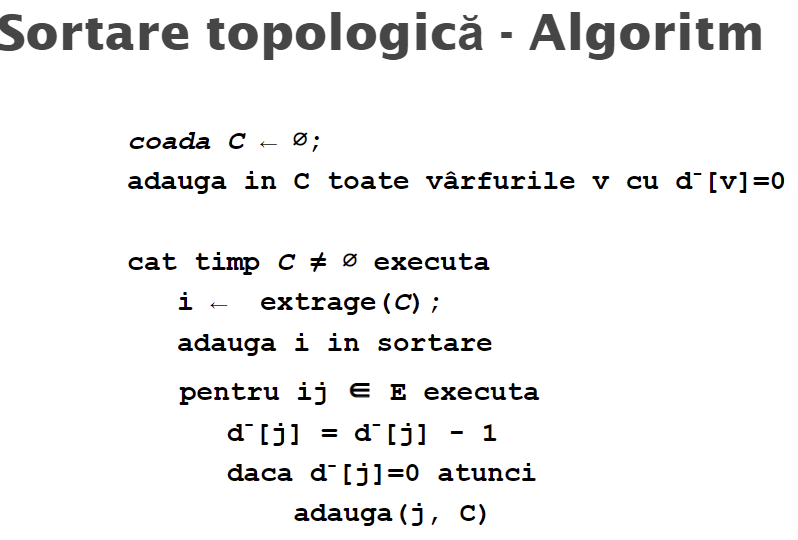


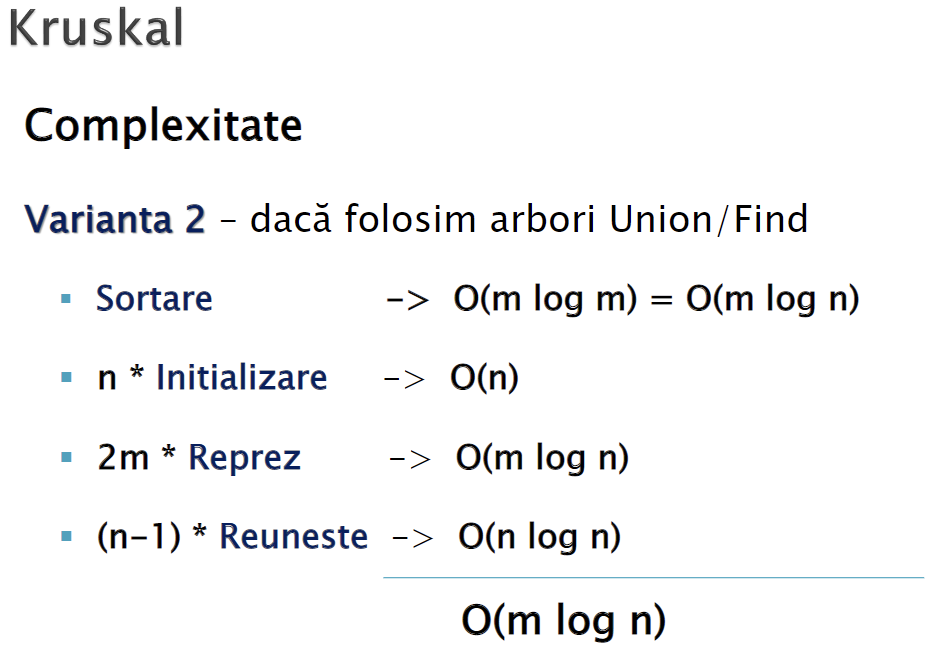


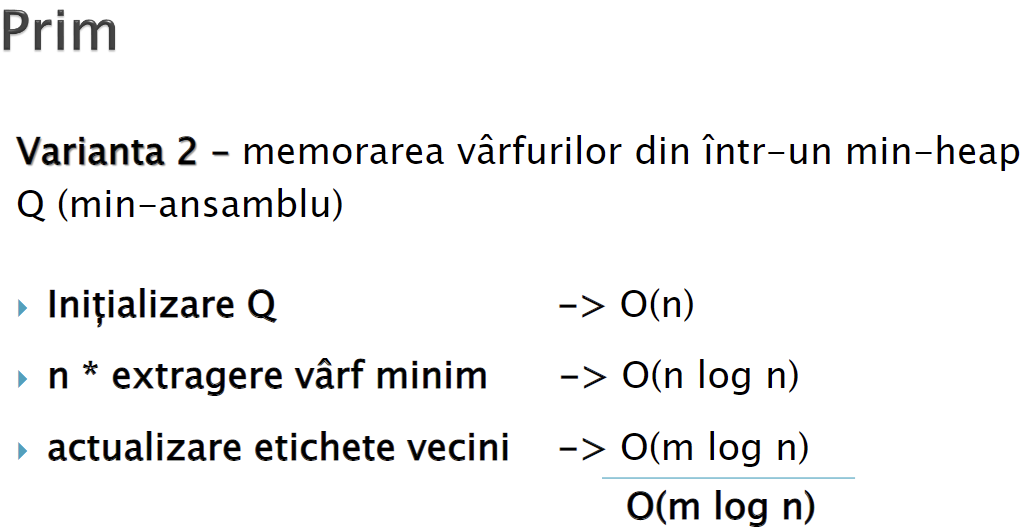












Dijkstra

