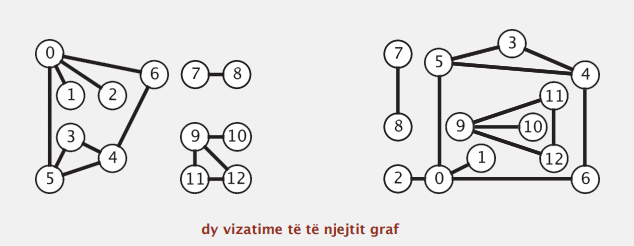
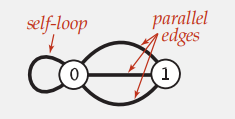
**UNDIRECTED GRAPHS**

Është më mirë që grafet fillimisht të viztohen e pastaj të programohen përshkak se e rritin intuitën për të mos gabuar.

Dy grafe mund të ken pamje të ndryshme por të jenë të njëjta :



Kulmet emërtohen me vlera integer ,prej 0 deri në V-1. (V-VertexKulmi , E-EdgeSegment). Me anë të symbolTables konverojmë emrin normal të kulemve në vlera integer.

Ekzistojnë dy **anomali** në grafe : **self-loop** dhe **paralel edges**. (Anomali : diqka që del jasht rregullave) 

**API-**Application Programme Interface : shërben për të shkruajtur një algoritëm në një program në mënyr që edhe programet tjera të munden ta lexojnë algoritmin.

Graph API:

public class Graph

Graph(int V) – Krijon një graf të zbrazët me V kulme integer

Graph(In in) – Krijon një graf të zbrazët me kulme të mara nga inputi

void addEdge(int V,int W) – Shton një segment që fillon nga kulmi V dhe shkon tek kulmi W

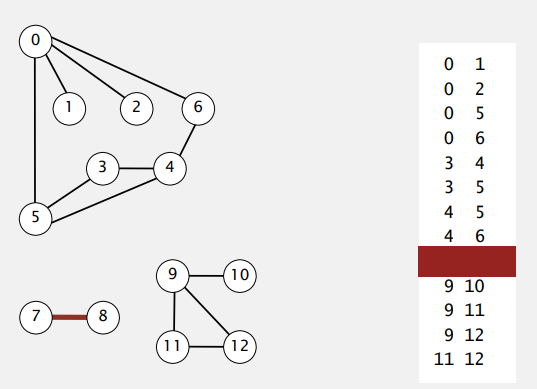
Iterable<Integer> adj(int V) – **adj**-adjacent(fqinjë) gjen kulmet e ngjitura me kulmin V

Int V() – Tregon numrin e Kulmeve të grafit (Vertex)

Int E() – Tregon numrin e segmenteve të grafit (Edge)

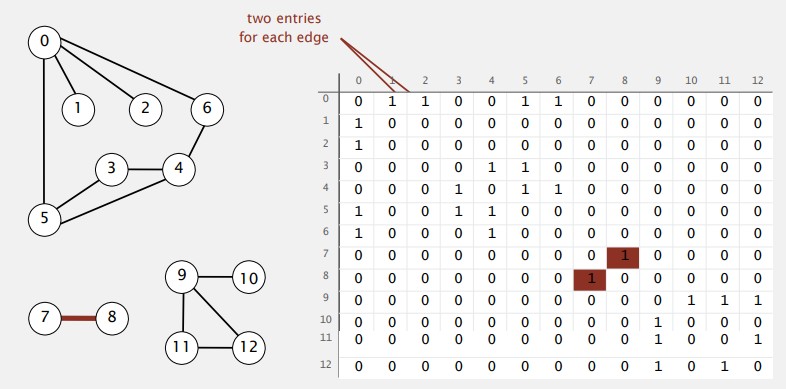
Paraqitja e **Set-of-Edges** graph (bashkësi e segmenteve)

Mirëmbajtia bëhet me anë të linked-list ose array

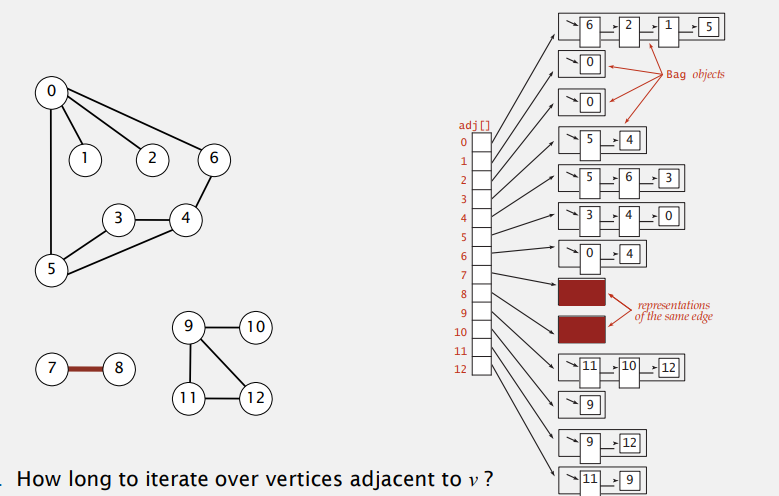


Paraqitja me **Adjancency-Matrix** graph

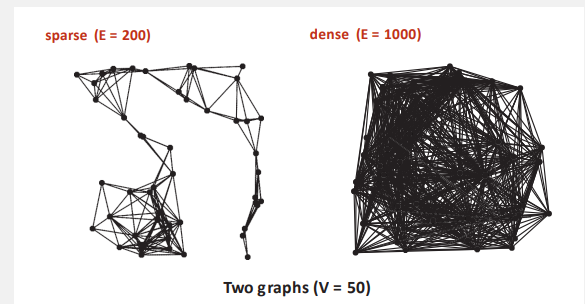
Është shumë e shpejtë për kërkime , mangësi e ksaj mënyre të paraqitje janë : hapsira e madhe që përdor me të dhëna të pa nevojshme dhe double data, nuk lejon parallel edges. Mirëmbajtia bëhet me array dy dimensinonale me vlera boolean.



Paraqitja me **Adjancency-list graf** graph

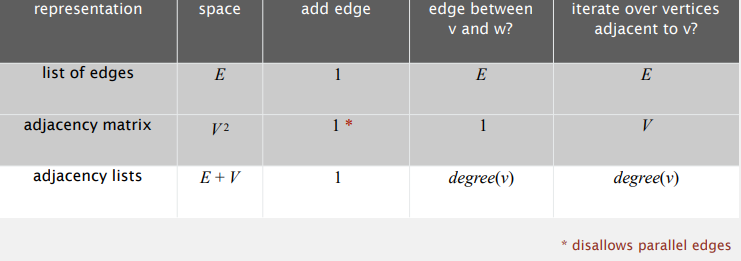
Mirëmbajtia bëhet me array of lists , mangësi e këti lloji të paraqitjes së grafeve është double data mirpo përdor më pak hapsirë se Matricat përshkak se nuk kemi 0 të pa nevojshe. 

Në praktitë grafet kanë tendencë të jenë sprase(të rralla) , edhe pse grafet dense(të dendura) janë më të shpejta , kjo përshkak se grafet sprase kushtojnë më lirë se ato dense.



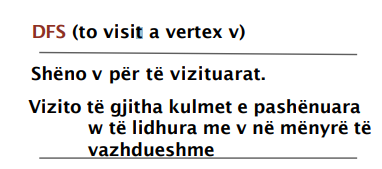
Sprase – numër i madh i kulmeve , mesatare më e ultë e deggre të kulemeve (vertex).

Tri mënyrat e parqitjes së grafeve dhe reprezentimi i tyre :



Siç shihet për sa i përket hapsirës **List of Edges** është më e mira, për të shtuar semgente më e keqja është **matrica** spse nëse kemi anomali nuk mund ta paraqesim, për të gjetur numrin e segmenteve(edges) në mes kulmeve (Vertex) më e shpejta është **matrix**(matrica).

**DFS-** Depth First Search : Është një algoritëm për kërkim i cili implementohet me stacks , qëllimi i ktij algoritmi është të përshkoj grafin në mënyrë sistematike, kërkimi fillon nga një kul që vet ne e përcaktojmë i cili quhet **Source Vertex**  ( kulm i burimit) , nëse nuk kemi një të till të caktuar e marin gjithomnë 0 si **source vertex**.



Aplikimet tipike të **DFS** : Per gjetjen e të gjitha kulmeve të lidhura me një source kulm(Vertex), Gjetja e një rruge në mes të dy kulmeve , flood fill ne photo shop.

Ideja e **DFS** : Vizito secilin kulm dhe pas vizitës të shenohet si i vizituar , nëse hasim në të vizituar kthehemi mbrapa duke kërkuar kulme të pa vizituara, kjo bëhet deri të arrijmë në fund.

**DFS** është më i mir për tu përdorur kur duam të gjej një rrug të cilën nuk kemi ide nga të fillojm të kërkojmë.

**BFS –** Breadth First Search : Është algoritëm për kërkim implementohet me anë të queue, qëllimi i këtij algoritmi është ti vizitoj të gjitha kulmet duke filluar që nga kulmi burimor.

**BFS** punon në këtë mënyrë : fshin prej queue një kulm dhe vendos në queue të gjitha kulmet të cilat janë të lidhura me kulmin e larguar dhe kjo përsëritet deri sa queue të mos jet e zbrazët kompletë.

