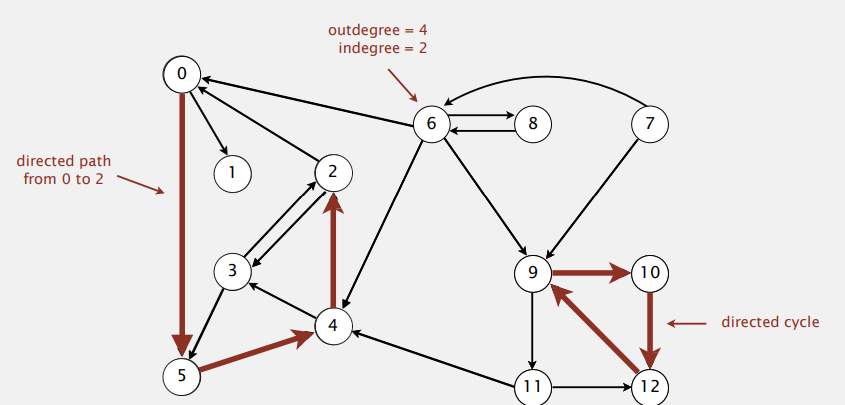
**Directed Graphs**

**Digraph** është bashkësi e kulmeve të lidhura në çfit me anë të **directed edges**. Shkalla (degree) e Digraph paraqet shumën e **outdegree** dhe **indegree** ,(segmenteve që dalin dhe atyre që hin në kulm).Një Digraph duhet ti ket të gjitha segmentet e orientuara. Zakonisht në praktit edhe Grafet edhe Digrafet paraqiten me Adjancy-List.



Digraph API :

public class Digraph

Digraph(int v) -Krijon një Digraf të zbrazët me V kulme(vertex)

Digraph(In in) -Krijon një Digraf të zbrazët me kulme nga input

void addEgde(Int v,int w) -Shton në Digraf një directed edge v -> w

Iterable<Integer> adj(int v) -Kulmet e lidhura me v

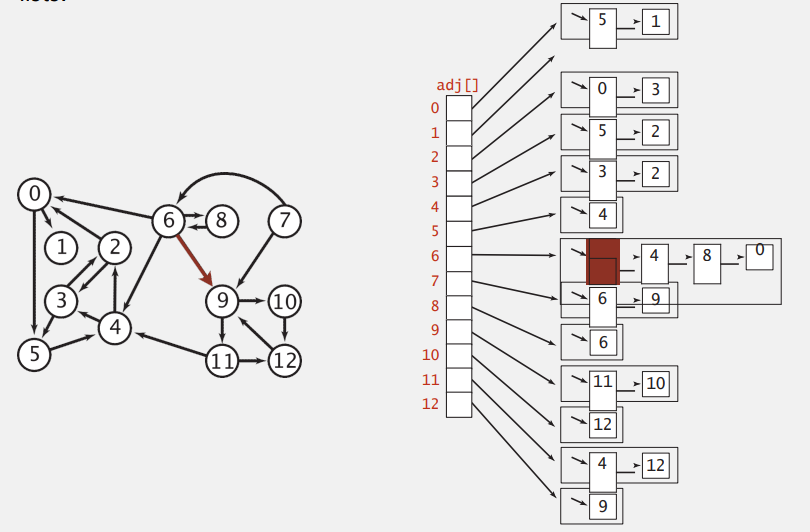
int V() -Tregon numrin e kulmeve

int E() -Tregon numrin e segmenteve

Digraph reserve() -Reverse of Digraph

String toString() -Reprezentimi i Digraph në String

Paraqitja e Digarph me **Adjacency-List :**  Mirëmbahet me anë të një array të listave, në array vendosen të gjitha kulmet e digrafit , ndërsa në secilin element të array janë të lidhura listat me të gjitha kulmet me të cilat kulmi elementi(kulmi) në atë pozit të array lidhet. Për ndallim nga Grafet tek Digrafet me paraqitjen **Adjacency-List** nuk shfaqet problem i double data.



Çdo undirected graph është një Digraph me segmente në të dyja drejtimet. DFS është një diraph algoritëm.

**Aplikimet e DFS tek Digraphs** : Reachability, path finding , topological sort, detektimi i directed cycle.

Kodi i cili është kod i pa aplikueshëm pra që nuk përdoret , quhet **Dead-Code**.

**Aplikimet e arrtishmërisë** (reachability) : **Program control-flow analystic -** Eliminimi i dead-code, detektimi i infinite-loop. **Mark-sweep garbage collector** (mark,shenje – sweep,fshij) i mark si root ato objetke që mund të qasen direct prej programit , ato që smund të qasen fshihen nga grabage collector sweep.

**Topological Sort :** Është algoritem për sortim që bën **precedence sechuling**(Kufizime të përparësisë). Për të funksionuar ky algoritëm nuk lejohet të ket cikle në graf. Ka dy rregulla- run multi-source DFS , returns kulmet në Reverse PostOrder .

**DAG** – Directed Acyclic Graph ( Digraf pa cikle).

**Conected Components :** Janë kulmet të cilat kan rrugë që i lidhin, ndërsa **Strongly CC** janë kulmet ku ka rrug që i mundësojnë të dy kulmeve të arrinjë njëri tjetrin.

**Kosaraju-Sharir** : Është algoritëm që mundëson të gjejmë **Strongly Conected Components.** Ka dy hapa : 1.DFS ne riverse graph , 2.DFS në grafin origjinal bazuar në Topological Sort values.