ADT Graf

- **ADT Graph** er i JDSL en positional container hvor posisjonene er noder og kanter i grafen.
- Metoder arvet fra Positional Container:
  - size() Antall kanter pluss noder.
  - isEmpty()
  - elements()
  - positions()
  - swap()
  - replaceElement()

Notasjon: Graph G; Vertices v, w; Edge e; Object o

- Metoder arvet fra InspectableGraph
  - numVertices()
    Return the number of vertices of G.
  - numEdges()
    Return the number of edges of G.
  - vertices() Return an enumeration of the vertices of G.
  - edges() Return enumeration of the edges of G.

Fra InspectableGraph

- directedEdges()
  Return an enumeration of all directed edges in G.
- undirectedEdges()
  Return an enumeration of all undirected edges in G.
- incidentEdges(v)
  Return an enumeration of all edges incident on v.
- inIncidentEdges(v)
  Return an enumeration of all the incoming edges to v.
- outIncidentEdges(v)
  Return an enumeration of all the outgoing edges from v.
- opposite(v, e)
  Return an endpoint of e distinct from v
- degree(v)
  Return the degree of v.
- inDegree(v)
  Return the in-degree of v.
- outDegree(v)
  Return the out-degree of v.

Fra InspectableGraph

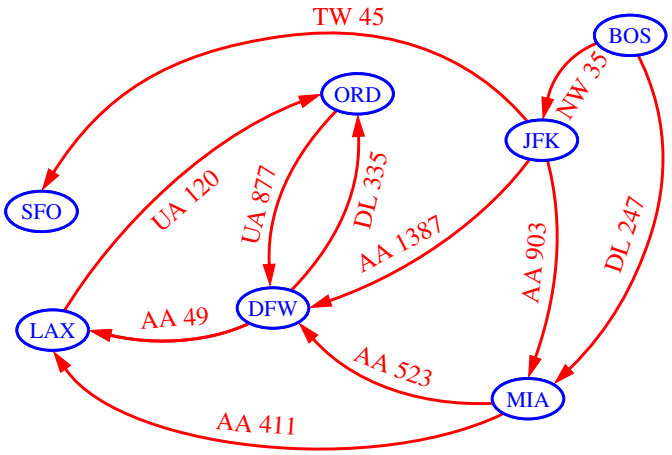
- adjacentVertices(v)
  Return an enumeration of the vertices adjacent to v.
- inAdjacentVertices(v)
  Return an enumeration of the vertices adjacent to v along incoming edges.
- outAdjacentVertices(v)
  Return an enumeration of the vertices adjacent to v along outgoing edges.
- areAdjacent(v,w)
  Return whether vertices v and w are adjacent.
- endVertices(e)
  Return an array of size 2 storing the end vertices of e.
- origin(e)
  Return the end vertex from which e leaves.
- destination(e)
  Return the end vertex at which e arrives.
- isDirected(e)
  Return true iff e is directed.

Oppdateringsmetoder i Graph:

- makeUndirected(e)
  Set e to be an undirected edge.
- reverseDirection(e)
  Switch the origin and destination vertices of e.
- setDirectionFrom(e, v)
  Sets the direction of e away from v
- setDirectionTo(e, v)
  Sets the direction of e toward v, one of its end vertices.
- insertEdge(v, w, o)
  Insert and return an undirected edge between v and w, storing o at this position.
- insertDirectedEdge(v, w, o)
  Insert and return a directed edge between v and w, storing o at this position.
- insertVertex(o)
  Insert and return a new (isolated) vertex storing o at this position.
- removeEdge(e)
  Remove edge e.

Implementasjon av ADT Graph

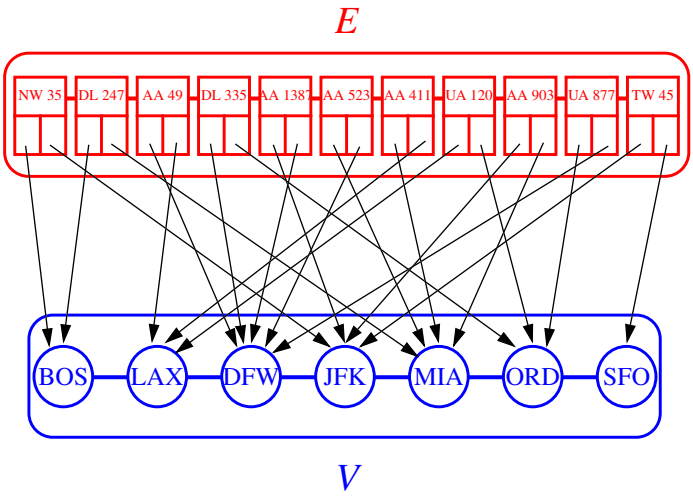
- Hvordan representere en graf?
- Vi lagrer noder og kanter i to containere, og hver kant peker til nodene den knytter sammen.



- I tillegg: 1)Kantliste, 2)Naboliste, 3)Nabomatrise

KantListe

- **Kantliste**-strukturen lagrer noder og kanter som usorterte sekvenser.
- Lett å implementere.
- Det å finne alle kanter tilhørende en node tar lang tid: må gå gjennom alle kanter.

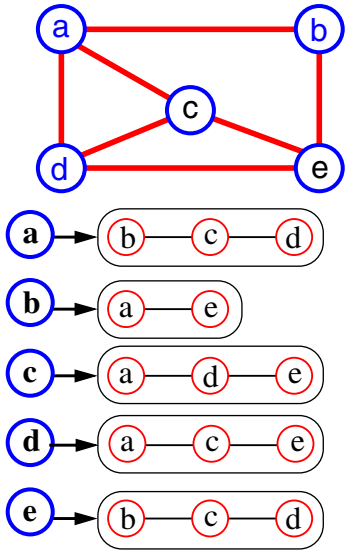


Kjøretid for Kantliste-operasjoner

Operasjon	Tid
size, isEmpty, replaceElement, swap	O(1)
numVertices, numEdges	O(1)
vertices	O(n)
edges, directedEdges, undirectedEdges	O(m)
elements, positions	O(n+m)
endVertices, opposite, origin, destination, isDirected	O(1)
incidentEdges, inIncidentEdges, outIncidentEdges, adjacentVertices, inAdjacentVertices, outAdjacentVertices, areAdjacent, degree, inDegree, outDegree	O(m)
insertVertex, insertEdge, insertDirectedEdge, removeEdge, makeUndirected, reverseDirection, setDirectionFrom, setDirectionTo	O(1)
removeVertex	O(m)

NaboListe (uten egne kantobjekter)

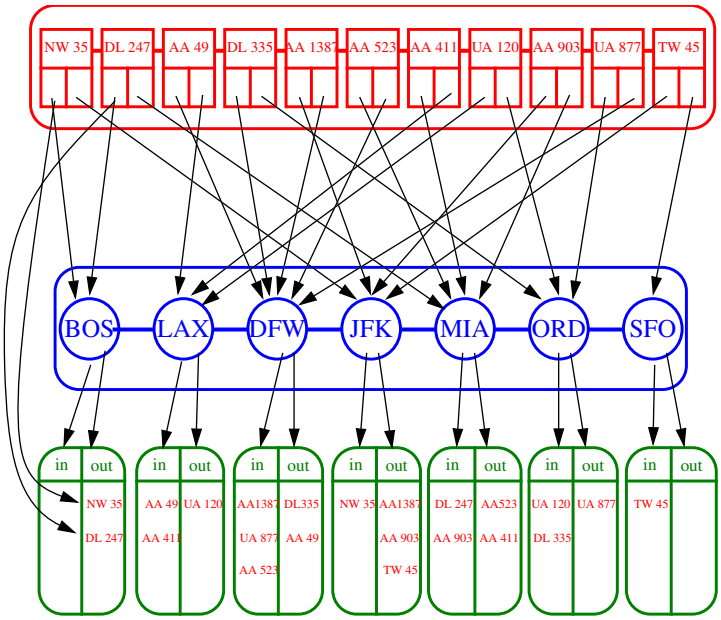
- naboliste for en node v:
  sekvens av noder som er nabo til v
- representer grafen ved egen naboliste for hver noder



- Minnebruk = O(**N** + $\sum$deg(**v**)) = O(**N** + **M**)

NaboListe
(med egne kantobjekter)

- **naboliste**-datastruktur utvider kantlistestrukturen med container over tilhørende kanter for hver node.

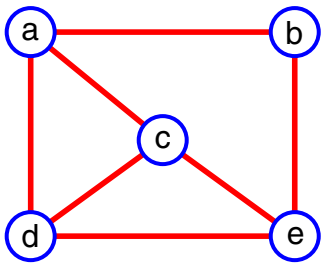


- Minnebruk O(n + m).

Kjøretid for NaboListe-Struktur

Operasjon	Tid
size, isEmpty, replaceElement, swap	O(1)
numVertices, numEdges	O(1)
vertices	O(n)
edges, directedEdges, undirectedEdges	O(m)
elements, positions	O(n+m)
endVertices, opposite, origin, destination, isDirected, degree, inDegree, outDegree	O(1)
incidentEdges(v), inIncidentEdges(v), outIncidentEdges(v), adjacentVertices(v), inAdjacentVertices(v), outAdjacentVertices(v)	O(deg(v))
areAdjacent(u, v)	O(min(deg(u), deg(v)))
insertVertex, insertEdge, insertDirectedEdge, removeEdge, makeUndirected, reverseDirection,	O(1)
removeVertex(v)	O(deg(v))

Nabomatrise
(uten egne kantobjekter)



	a	b	c	d	e
a	F	T	T	T	F
b	T	F	F	F	T
c	T	F	F	T	T
d	T	F	T	F	T
e	F	T	T	T	F

- matrise M for alle par av noder
- M[i,j] = True om kant (i,j) finnes i grafen.
- M[i,j] = False om det ikke er en kant (i,j) i grafen
- Minnebruk= O(N^2).

NaboMatrise
(med egne kantobjekter)

- Lagre selve kantobjektet i matrisen.

	0	1	2	3	4	5	6
0	Ø	Ø	NW 35	Ø	DL 247	Ø	Ø
1	Ø	Ø	Ø	AA 49	Ø	DL 335	Ø
2	Ø	AA 1387	Ø	Ø	AA 903	Ø	TW 45
3	Ø	Ø	Ø	Ø	Ø	UA 120	Ø
4	Ø	AA 523	Ø	AA 411	Ø	Ø	Ø
5	Ø	UA 877	Ø	Ø	Ø	Ø	Ø
6	Ø	Ø	Ø	Ø	Ø	Ø	Ø

BOS 0 | DFW 1 | JFK 2 | LAX 3 | MIA 4 | ORD 5 | SFO 6

- Minnebruk O(n^2 + m)

Kjøretid for nabomatrise-strukturen

Operasjon	Tid
size, isEmpty, replaceElement, swap	O(1)
numVertices, numEdges	O(1)
vertices	O(n)
edges, directedEdges, undirectedEdges	O(m)
elements, positions	O(n+m)
endVertices, opposite, origin, destination, isDirected, degree, inDegree, outDegree	O(1)
incidentEdges, inIncidentEdges, outIncidentEdges, adjacentVertices, inAdjacentVertices, outAdjacentVertices,	O(n)
areAdjacent	O(1)
insertEdge, insertDirectedEdge, removeEdge, makeUndirected, reverseDirection, setDirectionFrom, setDirectionTo	O(1)
insertVertex, removeVertex	$O(n^2)$