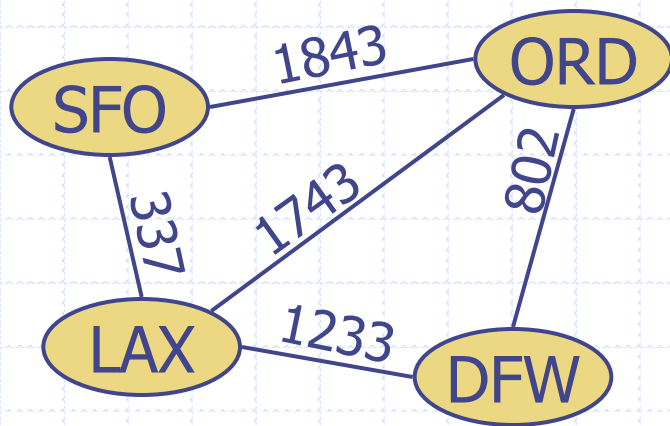
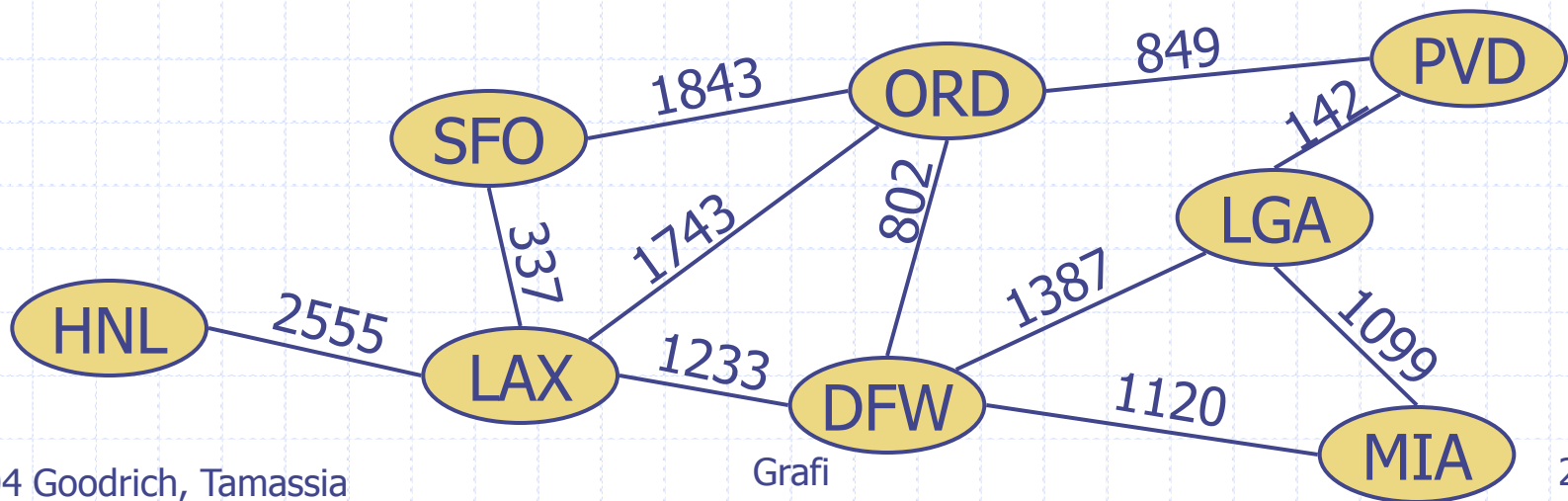


Grafi



Grafi (13.1)

- ◆ Un grafo è una coppia (V, E) , dove
 - V è un insieme di nodi, detti **vertici**
 - E è una collezione di coppie di vertici, dette **spigoli** (o **archi**)
 - Vertici e spigoli sono Position e contengono elementi
- ◆ Esempio:
 - Un vertice rappresenta un aeroporto e contiene il codice aeroportuale di tre lettere
 - Uno spigolo rappresenta una rotta di volo fra due aeroporti e contiene la lunghezza della rotta



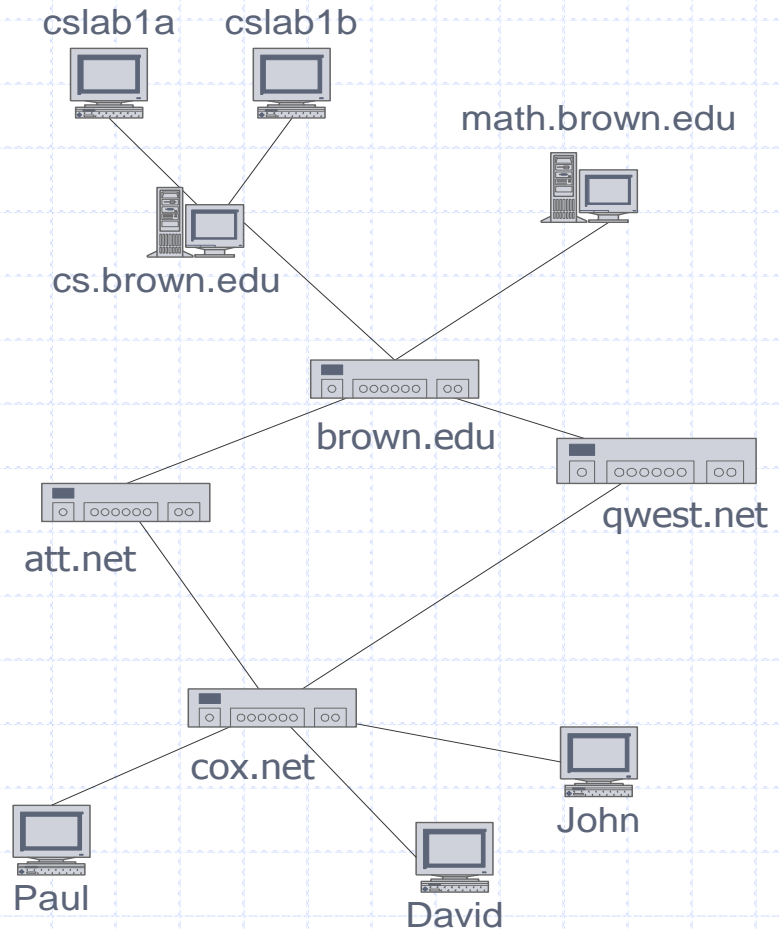
Tipi di spigolo

- ◆ Spigolo orientato (o diretto)
 - coppie ordinate di vertici (u,v)
 - il primo vertice u è l'origine
 - il secondo vertice v è la destinazione
 - ad esempio, un volo
- ◆ Spigolo non orientato (o non diretto)
 - coppia non ordinata di vertici (u,v)
 - ad esempio, una rotta
- ◆ Grafo diretto (od orientato)
 - tutti gli spigoli sono orientati
 - ad esempio, una rete di voli
- ◆ Grafo non diretto (o non orientato)
 - tutti gli spigoli sono non orientati
 - ad esempio, una rete di rotte aeree



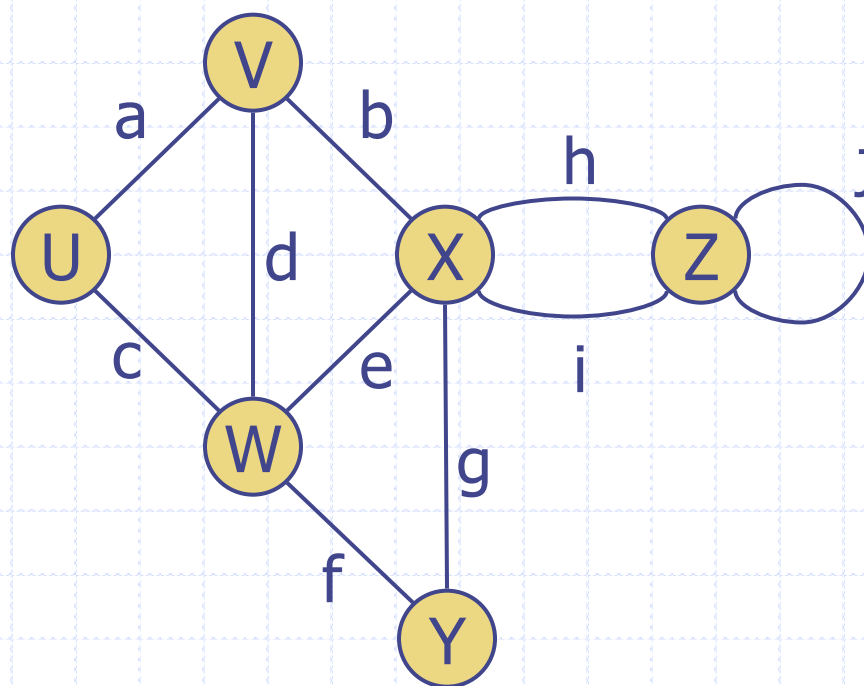
Applicazioni

- ◆ Circuiti elettronici
 - Circuito stampato
 - Circuiti integrati
- ◆ Reti di trasporto
 - Rete autostradale
 - Rete di voli
- ◆ Reti di computer
 - LAN
 - Internet
 - Web
- ◆ Basi di dati
 - Diagramma entità-relazione



Terminologia

- ◆ Vertici (o punti) terminali di uno spigolo
 - U e V sono i punti terminali di a
- ◆ Spigoli incidenti su un vertice
 - a, d e b sono incidenti su V
- ◆ Vertici adiacenti
 - U e V sono adiacenti
- ◆ Grado di un vertice
 - X ha grado 5
- ◆ Spigoli paralleli (o multipli)
 - h ed i sono spigoli paralleli
- ◆ Autoanelli (self-loop)
 - j è un self-loop



Terminologia (cont.)

◆ Percorso (path)

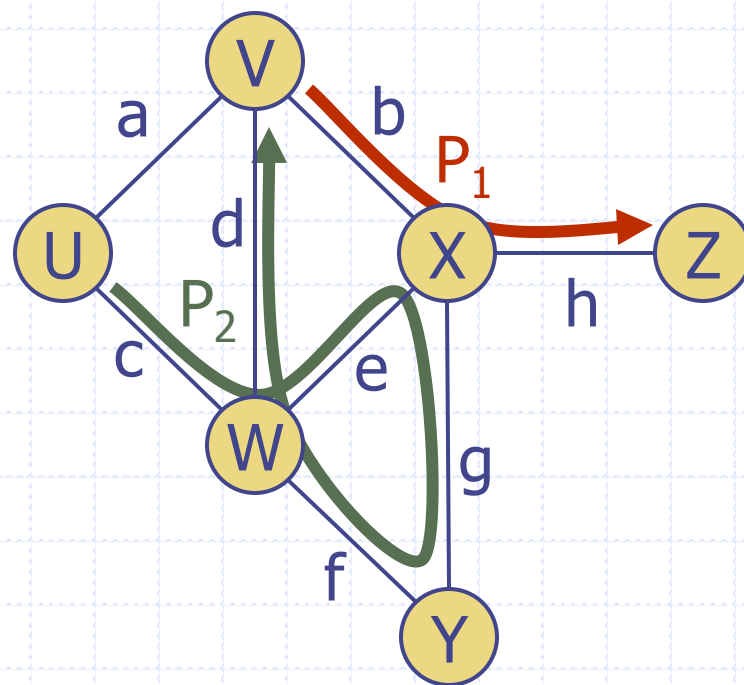
- sequenza di vertici alternati a a spigoli
- inizia con un vertice
- termina con un vertice
- ciascuno spigolo è preceduto e seguito dai punti terminali

◆ Percorso semplice

- percorso in cui tutti i vertici e spigoli sono distinti

◆ Esempi

- $P_1 = (V, b, X, h, Z)$ è un percorso semplice
- $P_2 = (U, c, W, e, X, g, Y, f, W, d, V)$ è un percorso non semplice



Terminologia (cont.)

◆ Ciclo

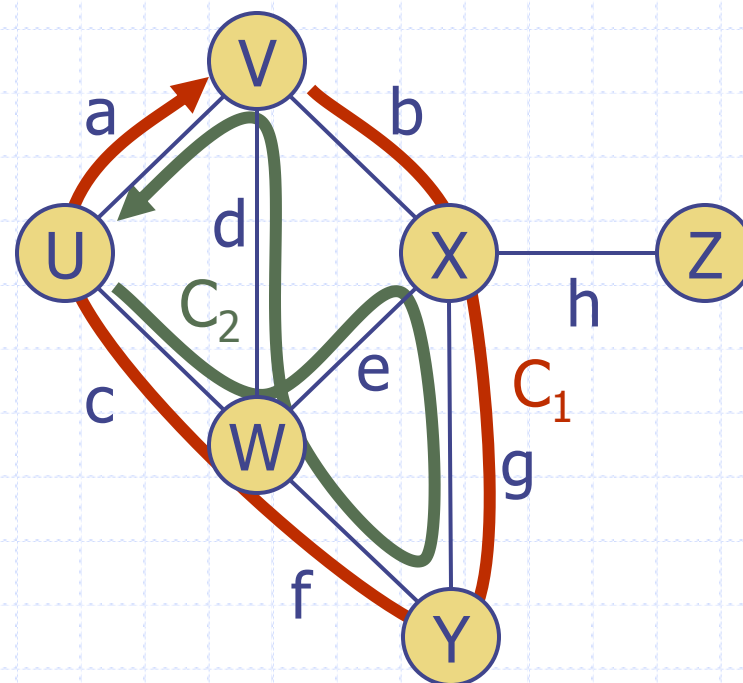
- sequenza circolare di vertici alternati a spigoli
- ciascuno spigolo è preceduto e seguito dai suoi punti terminali

◆ Ciclo semplice

- ciclo in cui tutti i vertici e spigoli sono distinti

◆ Esempi

- $C_1 = (V, b, X, g, Y, f, W, c, U, a, \downarrow)$ è un ciclo semplice
- $C_2 = (U, c, W, e, X, g, Y, f, W, d, V, a, \downarrow)$ è un ciclo non semplice



Proprietà

Proprietà 1

$$\sum_v \deg(v) = 2m$$

Prova: ciascuno spigolo è contato due volte

Proprietà 2

In un grafo non diretto e privo di autoanelli e di spigoli multipli

$$m \leq n(n-1)/2$$

Prova: ciascun vertice ha grado al più $(n-1)$

Che accade nel caso di grafi diretti?

Notazione

n

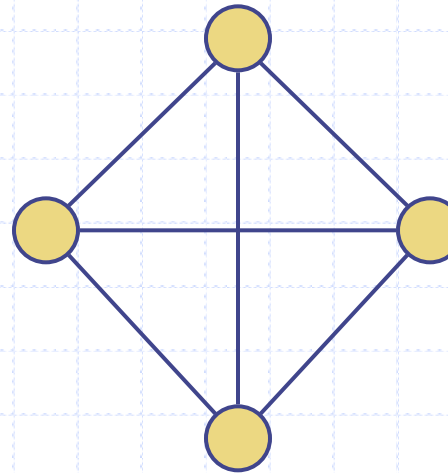
numero di vertici

m

numero di spigoli

$\deg(v)$

grado del vertice v



Esempio

■ $n = 4$

■ $m = 6$

■ $\deg(v) = 3$

Metodi principali del TdA Grafo

◆ Vertici e spigoli

- sono Position
- contengono elementi

◆ Metodi accessori

- **endVertices**(e): array con i due vertici terminali di e
- **opposite**(v , e): il vertice opposto di v , sullo spigolo e
- **areAdjacent**(v , w): vero se e solo se v e w sono adiacenti
- **replace**(v , x): sostituisce l'elemento nel vertice v con x
- **replace**(e , x): sostituisce l'elemento nello spigolo e con x

◆ Metodi modificatori

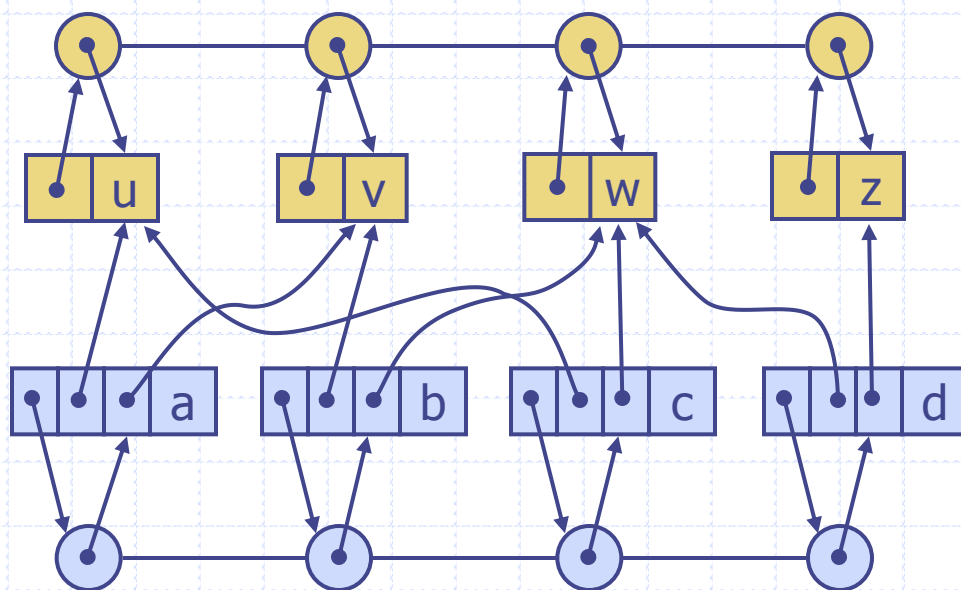
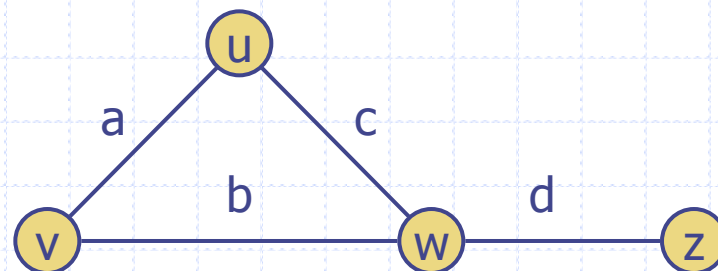
- **insertVertex**(o): inserisce un vertice contenente l'elemento o
- **insertEdge**(v , w , o): inserisce uno spigolo (v , w) contenente l'elemento o
- **removeVertex**(v): rimuove il vertice v (e gli spigoli incidenti)
- **removeEdge**(e): rimuove lo spigolo e

◆ Metodi iteratori

- **incidentEdges**(v): spigoli incidenti su v
- **vertices**(): tutti i vertici nel grafo
- **edges**(): tutti gli spigoli nel grafo

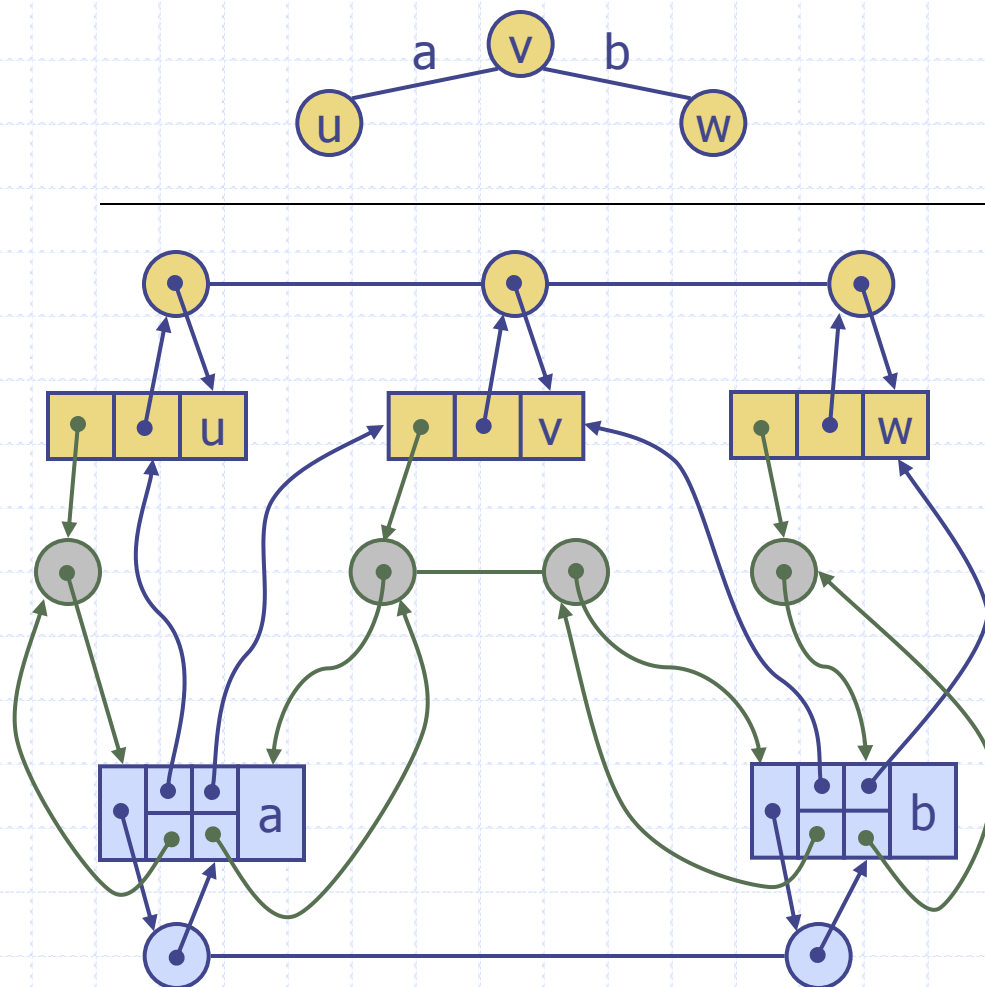
La lista degli spigoli (13.2.1)

- ◆ Oggetto Vertex
 - elemento
 - riferimento alla posizione nella sequenza di vertici
- ◆ Oggetto Edge
 - elemento
 - riferimento all'oggetto vertice origine
 - riferimento all'oggetto vertice destinazione
 - riferimento alla posizione nella sequenza di spigoli
- ◆ Sequenza di vertici
 - sequenza di oggetti Vertex
- ◆ Sequenza di spigoli
 - sequenza di oggetti Edge



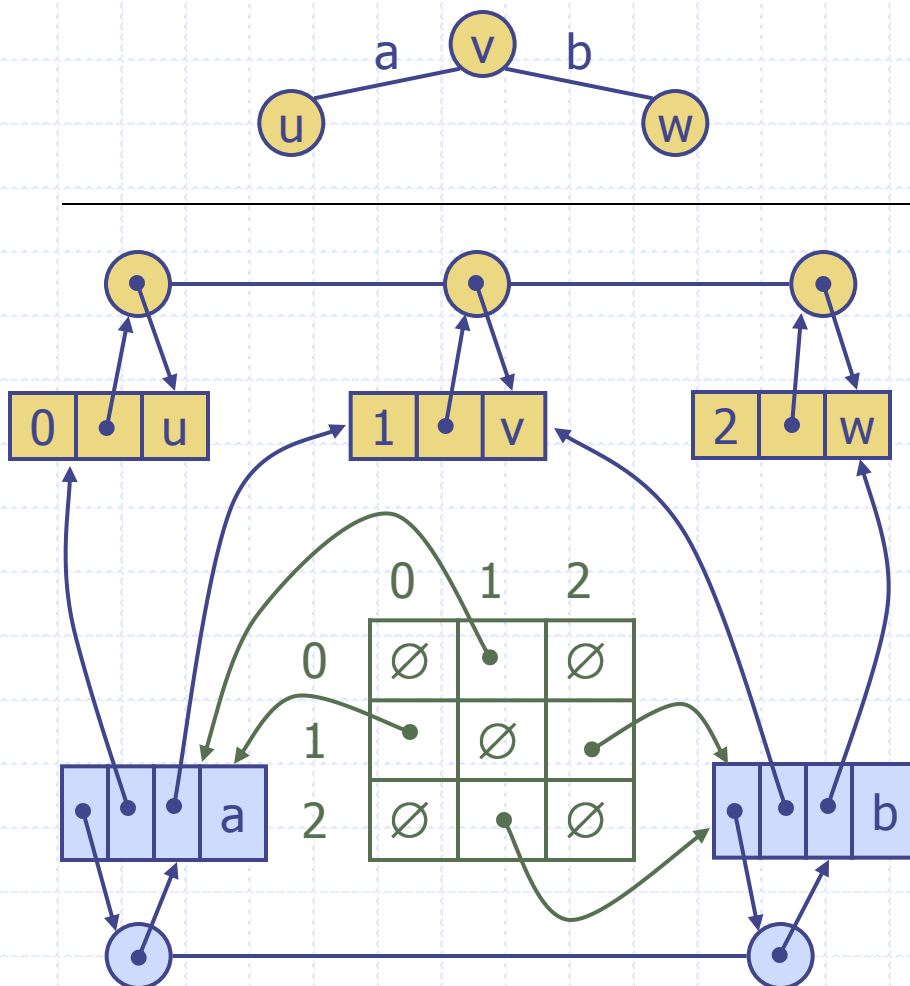
La lista delle adiacenze (13.2.2)

- ◆ Lista degli spigoli
- ◆ Sequenza di incidenze per ciascun vertice
 - sequenza di riferimenti agli oggetti spigolo associati agli spigoli incidenti il vertice
- ◆ Oggetti spigolo incrementati
 - riferimenti alle posizioni nelle sequenze di incidenza dei vertici terminali



Matrice delle adiacenze (13.2.3)

- ◆ Lista degli spigoli
- ◆ Oggetti vertice aumentati
 - ad ogni vertice viene associata una chiave intera (indice)
- ◆ Matrice delle adiacenze
 - Riferimenti all'oggetto spigolo per ciascuna coppia di vertici adiacenti
 - null per coppie di vertici non adiacenti
- ◆ La versione tradizionale prevedeva solamente 1 o 0 (spigolo presente o assente)



Prestazioni asintotiche

| | | | |
|---|------------------|-----------------------------|----------------------|
| <ul style="list-style-type: none"> ◆ n vertici, m spigoli ◆ no spigoli multipli ◆ no autoanelli ◆ Costi in O-grande | Lista spigoli | Lista adiacenze | Matrice adiacenze |
| Space | $n + m$ | $n + m$ | n^2 |
| incidentEdges(v) | m | deg(v) | n |
| areAdjacent (v, w) | m | min(deg(v), deg(w)) | 1 |
| insertVertex(o) | 1 | 1 | n^2 |
| insertEdge(v, w, o) | 1 | 1 | 1 |
| removeVertex(v) | m | deg(v) | n^2 |
| removeEdge(e) | 1 | 1 | 1 |