Lecture
8_9.1

Graphs: Introduction & definitions

cini
Cybersecurity
National Lab

Paolo PRINETTO
Politecnico di Torino (Italy)
Univ. of Illinois at Chicago, IL (USA)
CINI Cybersecurity Nat. Lab. (Italy)

Paolo.Prinetto@polito.it

www.consorzio-cini.it
www.comitato-girotondo.org

License Information

This work is licensed under the Creative Commons BY-NC License

To view a copy of the license, visit:
http://creativecommons.org/licenses/by-nc/3.0/legalcode

Rel. 7.04.2019

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

Rel. 7.04.2019

Goal

– **This lecture aims at presenting the *Graph* ADT and its implementation.**

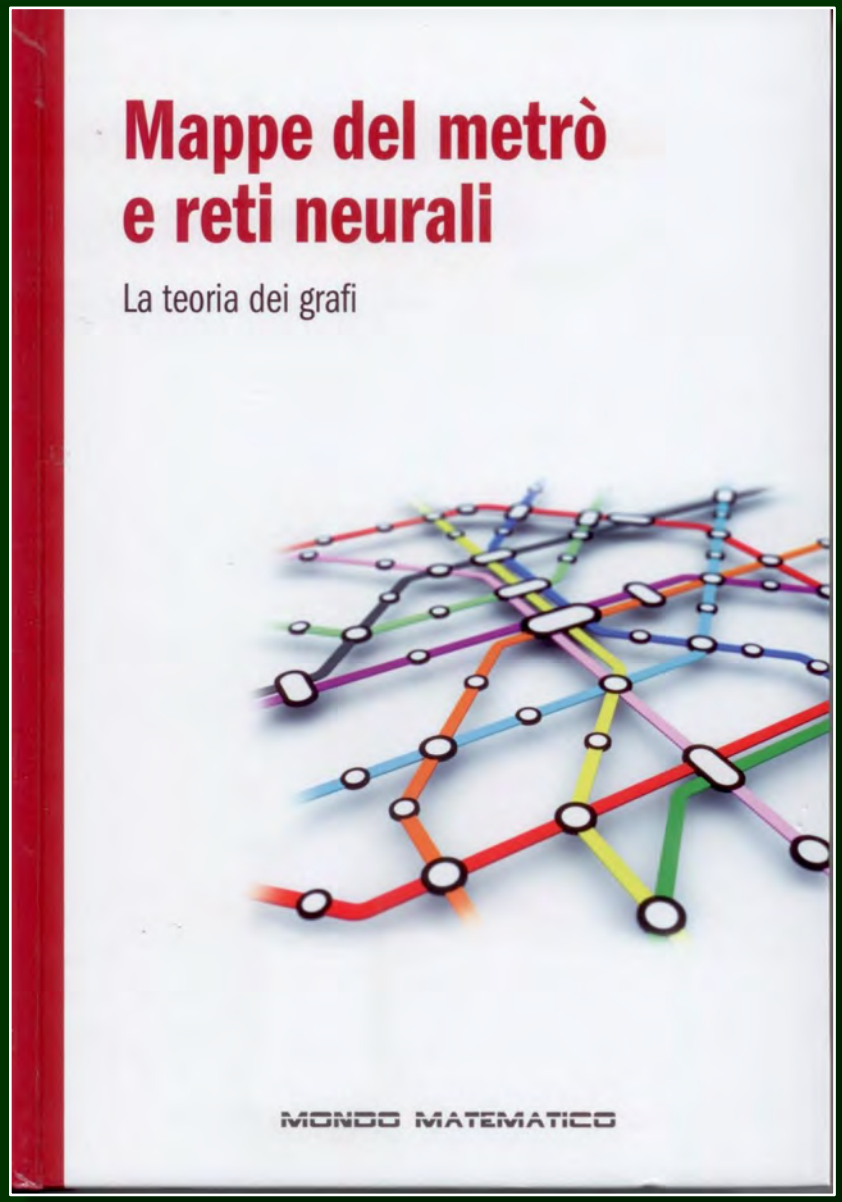
Prerequisites

- **Lectures:**
  - **8_1.1 Pointers & Dynamic Memory**
  - **8_2.1 Introduction to ADT**

Rel. 7.04.2019

Further readings

- **Students interested in a deeper look at the covered topics can refer, for instance, to the books listed at the end of the lecture.**



Rel. 7.04.2019

Outline

- Introduzione
- Definizioni varie
- Rappresentazione di un grafo

Rel. 7.04.2019

Outline

- Introduzione
- Definizioni varie
- Rappresentazione di un grafo

Rel. 7.04.2019

Introduzione

- **La grande branca della Teoria dei Grafi è stata motivata dalla Teoria dei Giochi e dalla matematica ricreativa.**
- **In generale, si usano i grafi in due situazioni.**

Rel. 7.04.2019

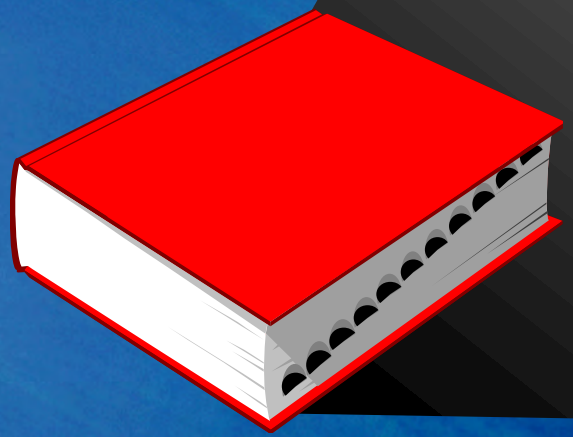
Utilizzo dei grafi

- Primo, siccome un grafo è un modo molto appropriato e naturale di rappresentare delle relazioni fra degli oggetti, rappresentiamo gli oggetti con vertici e le relezioni fra di essi con delle linee.
- Esempio di tali applicazioni sono: molecole chimiche, colorazione di mappe, signal-flow graphs, i ponti di Konigsberg, i labirinti, …

Rel. 7.04.2019

Utilizzo dei grafi

- **Secondo, prendiamo il grafo con un modello matematico, risolviamo il problema di teoria dei grafi adatto, e in fine interpretiamo la soluzione in termini del problema originale.**



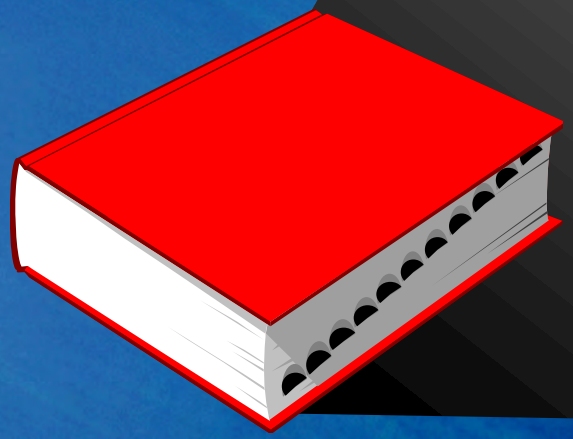
Grafo

Si definisce grafo una coppia ordinata (V, E) di insiemi:

- ***V è un insieme finito e non vuoto di oggetti, denominati nodi o vertici***
- ***E è un insieme di archi o spigoli (edges)***

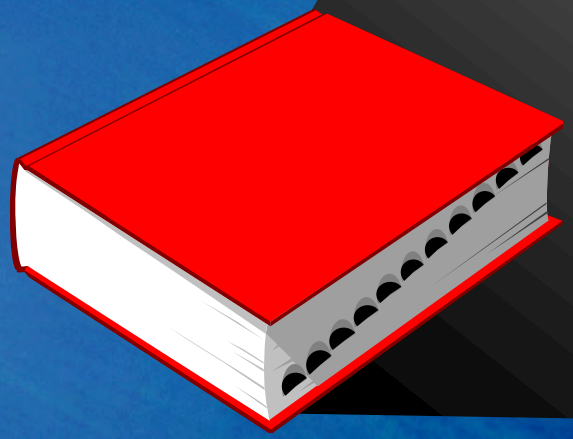
Arco

Un arco è una coppia di nodi, che risultano collegati dall'arco stesso.



Rel. 7.04.2019

Operatore | |



Nel seguito indicheremo con:

- ***|V| il numero dei vertici***
- ***|E| il numero degli archi.***

Rappresentazione Grafica di un Grafo

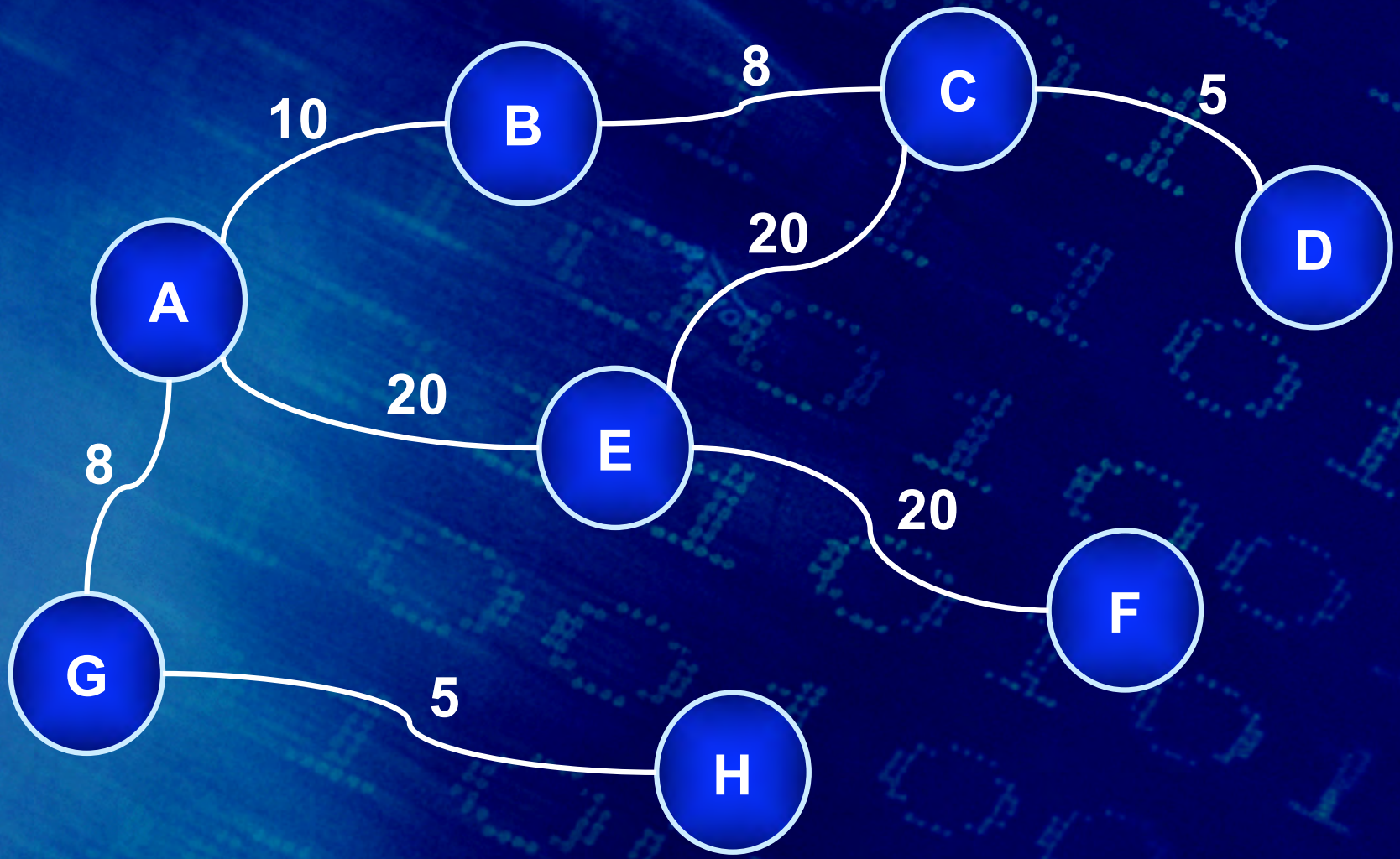
Per disegnare un grafo si rappresentano usualmente i nodi tramite punti o cerchi e gli archi come linee che collegano i nodi.

Esempio: mappa dei collegamenti aerei tra varie città

Linee telefoniche

- **Una compagnia telefonica gestisce le centrali di commutazione con linee aventi differenti larghezze di banda.**
- **Per rappresentare la rete, si può usare un grafo in cui i nodi sono le centrali di commutazione, gli archi le linee di connessione, e i *pesi* le rispettive larghezze di banda.**

Linee telefoniche



Rel. 7.04.2019

Linee telefoniche

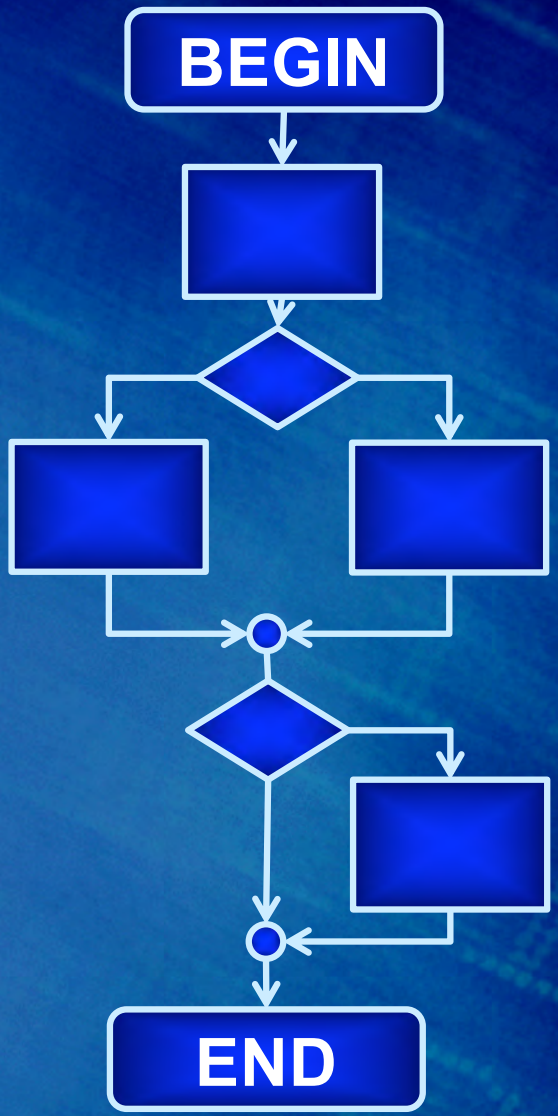
- **Problemi:**
  - **Il grafo è connesso? Se non lo è, ci saranno utenti isolati.**
  - **Quali sono le linee critiche (cioè quelle che, se rimosse, renderanno il grafo non connesso)?**
  - **Quali sono le centrali critiche?**

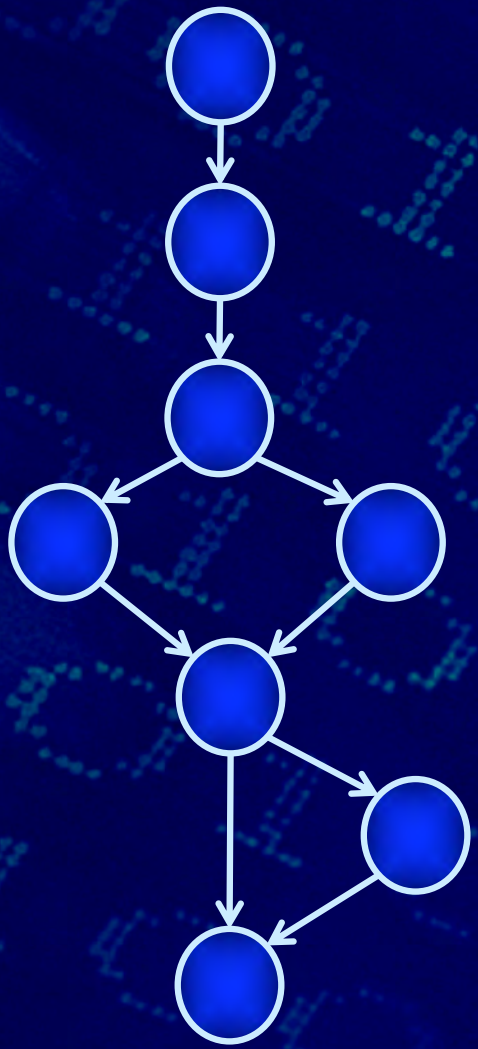
Flow chart

- In software engineering, la struttura del codice è analizzata utilizzando dei grafi:
  - I nodi sono istruzioni sequenziali
  - Gli archi sono i possibili flussi di esecuzione
  - Possiamo analizzare il comportamento del programma visitando il grafo

Rel. 7.04.2019

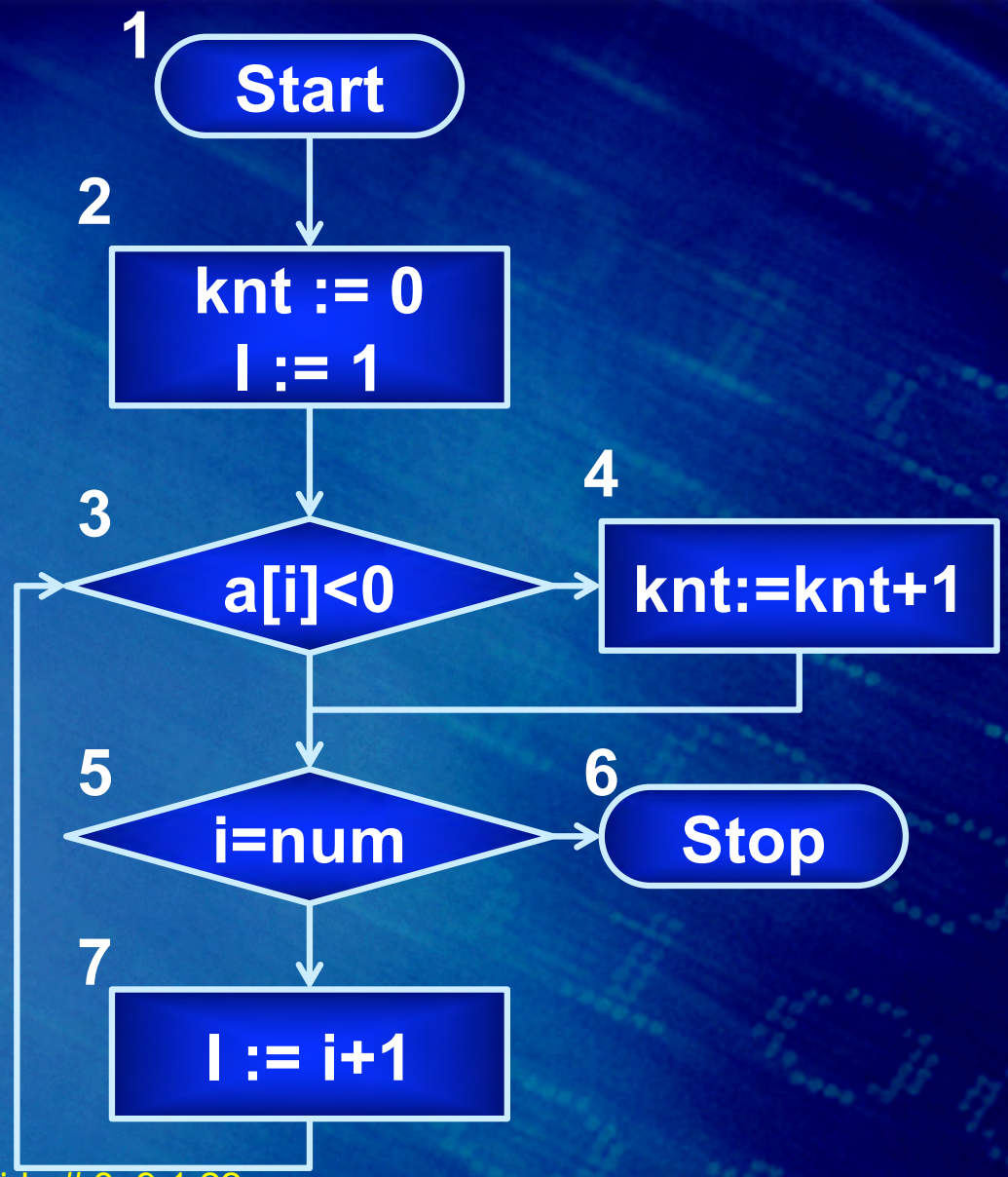
Flow chart

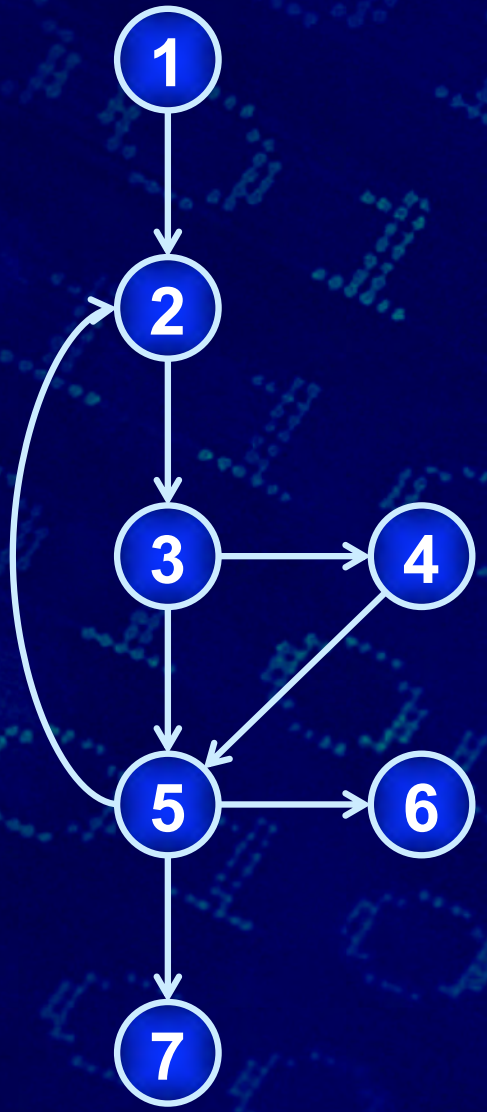




Rel. 7.04.2019

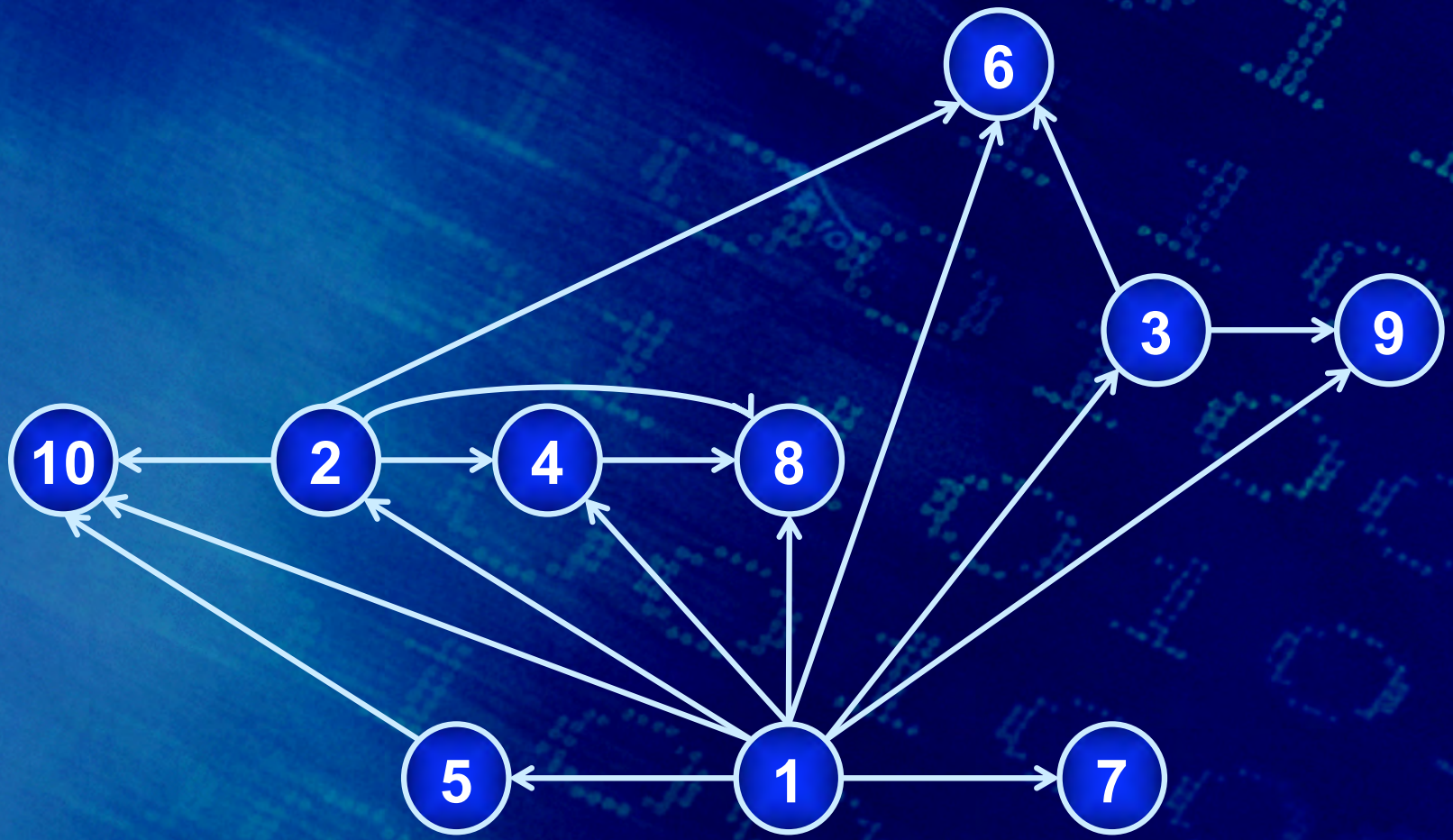
Esempio: flusso di controllo in un flowchart





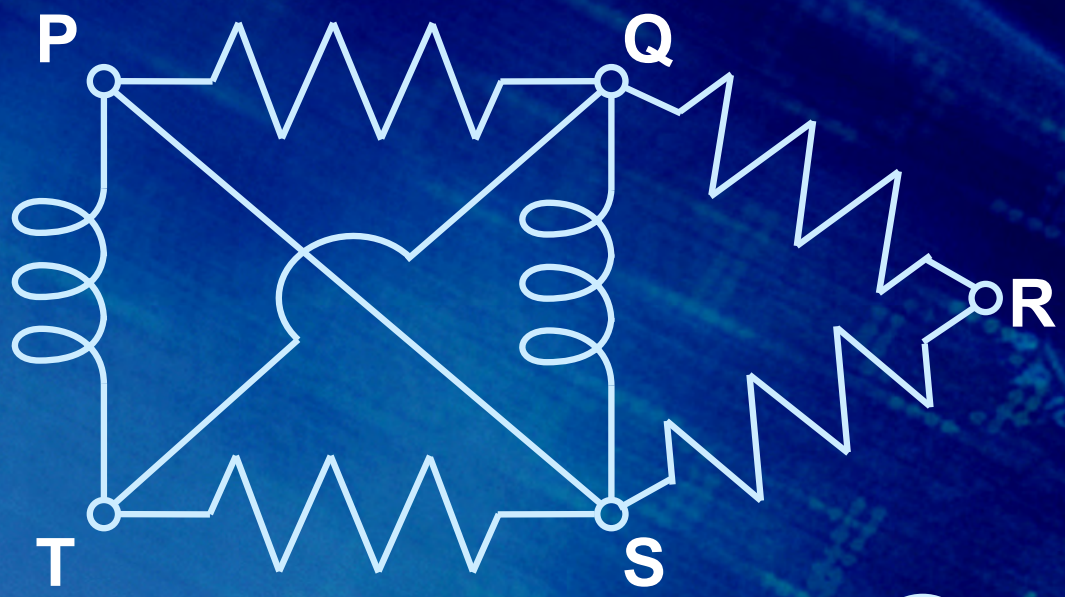
Esempio: relazione binaria

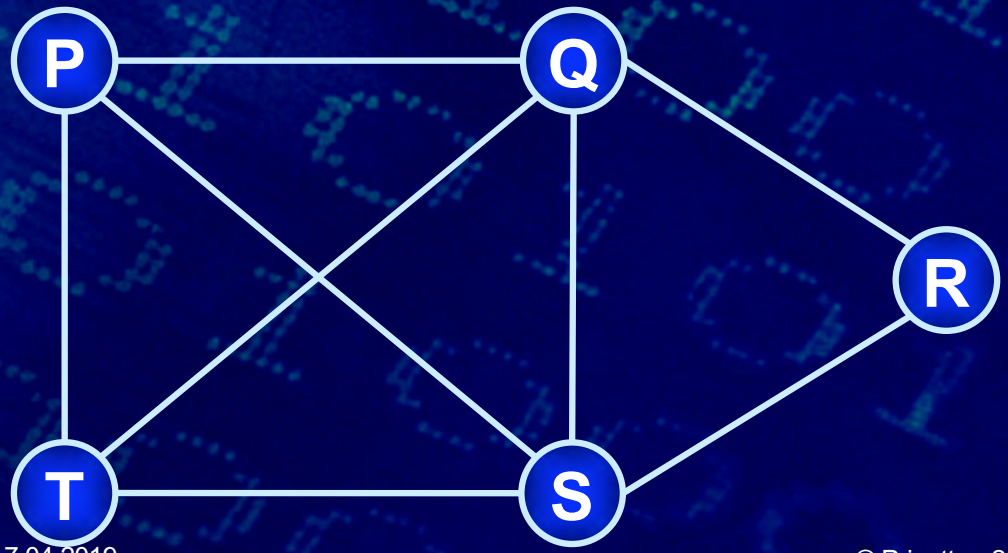
$x \neq y$ e x divide y



Rel. 7.04.2019

Esempio: circuiti elettrici



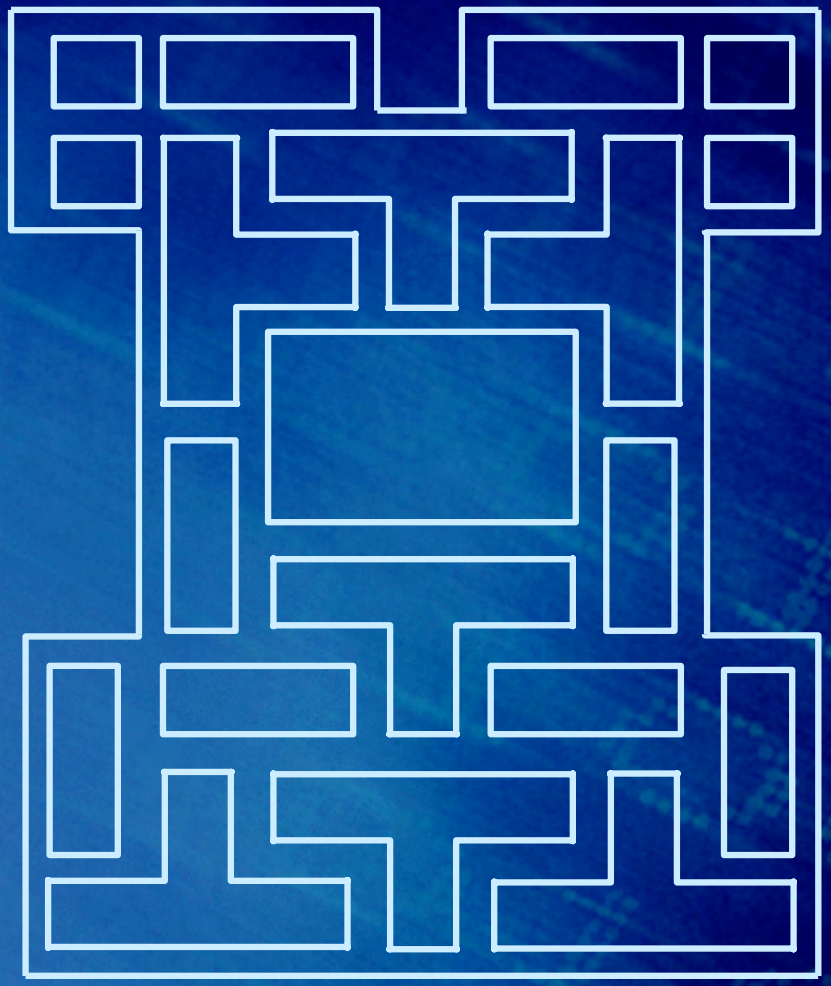


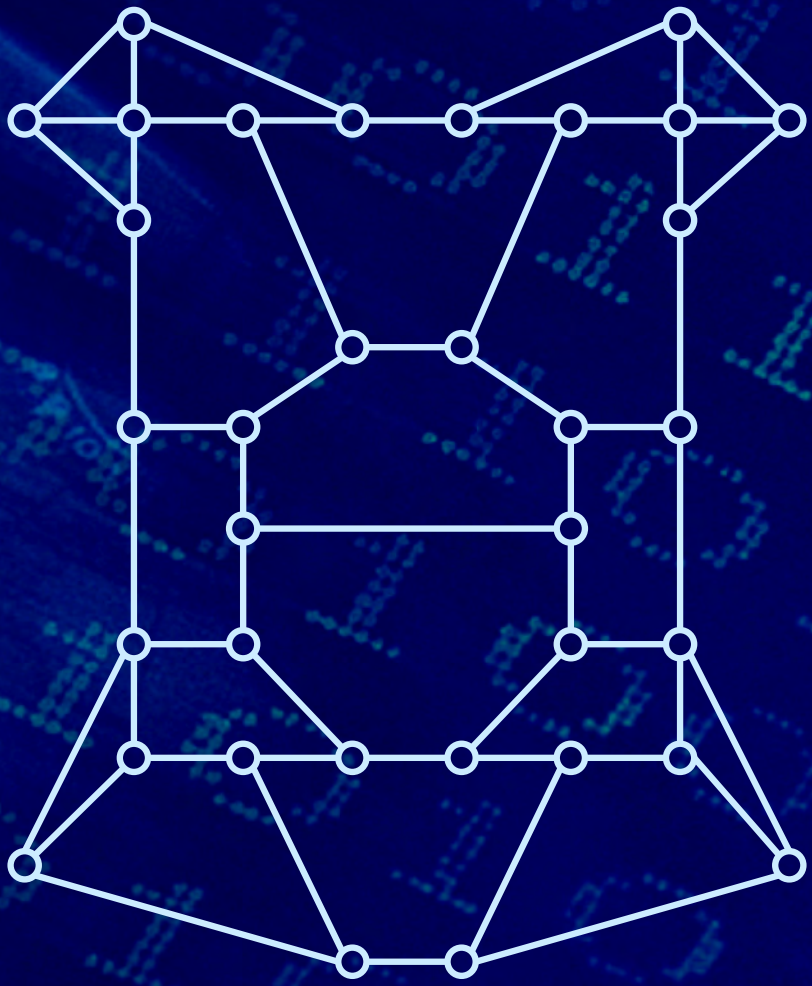
Rel. 7.04.2019

Esempio: mappa dei percorsi stradali con le relative capacità di traffico

Rel. 7.04.2019

Esempio: rappresentazione di labirinti





Outline

- Introduzione
- Definizioni varie
  - Grafi orientati e non orientati
  - Multigrafi
  - Grafi completi
  - Grafi regolati
  - Grafi pesati
  - Sottografi
  - Grafi bipartiti
- Rappresentazione di un grafo

Rel. 7.04.2019

Grafi non orientati

Quando gli archi non sono orientati, ossia quando non hanno una direzione, il grafo si dice **non orientato** o **non diretto (undirected graph)**.

Ogni arco e in un grafo non orientato è rappresentabile come una coppia **non ordinata** di vertici:

$$e = \{ v_1, v_2 \}$$

Rel. 7.04.2019

Grafi non orientati

Ne consegue che $\{ v_1, v_2 \}$ e $\{ v_2, v_1 \}$ rappresentano lo stesso arco.

I vertici v_1 e v_2 si dicono **adiacenti** e l'arco e è **incidente** sui vertici v_1 e v_2.

Cappi

Sono normalmente ammessi archi in cui i due vertici siano coincidenti:

$$e = \{ v_1, v_2 \}$$

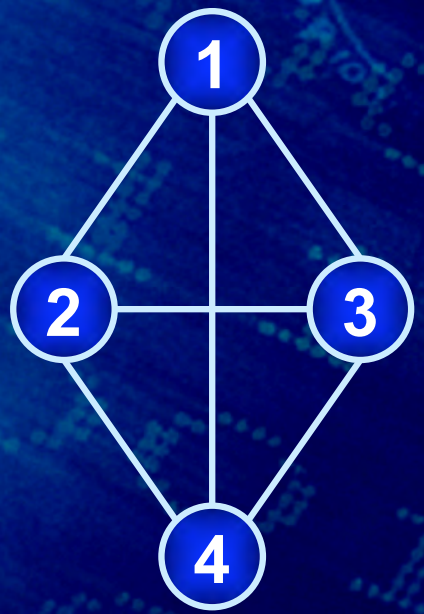
o

$$e = \{ v_2, v_1 \}$$

In questo caso l'arco si dice cappio (self loop).

Esempi di grafi non orientati

- $V(G_1) = \{ 1, 2, 3, 4 \}$
- $E(G_1) = \{ \{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\} \}$



G_1

Rel. 7.04.2019

Esempi di grafi non orientati

$V(G_2) = \{ 1, 2, 3, 4, 5, 6, 7 \}$

$E(G_2) = \{ \{1,2\}, \{1,3\}, \{2,4\}, \{2,5\}, \{3,6\}, \{3,7\} \}$

G_2

Grafi orientati

Se gli archi hanno una direzione, il grafo si dice orientato o diretto (directed graph, o digraph).

Ogni arco e in un grafo orientato è rappresentabile come una coppia orientata di vertici:

$$e = (v_1, v_2)$$

Grafi orientati

In questo caso si dice che:

- v_1 è la coda di e
- v_2 è la testa di e
- v_1 è un predecessore di v_2
- v_2 è un successore di v_1
- v_1 è adiacente a v_2
- v_2 è adiacente da v_1
- e è incidente da v_1
- e è incidente in v_2.

Esempio di grafo orientato

$V(G_3) = \{ 1, 2, 3 \}$

$E(G_3) = \{ (1,2), (2,1), (2,3) \}$

G_3

Multigrafi

- Poiché *E* è, per definizione, un insieme, non può contenere elementi uguali
- Ne consegue che il grafo non può contenere archi multipli
- Nel caso in cui tale limitazione venga rimossa, la struttura dati ottenuta viene chiamata **multigrafo (multigraph)**.

Rel. 7.04.2019

Grafi semplici

Un grafo viene definito semplice se non contiene cappi.

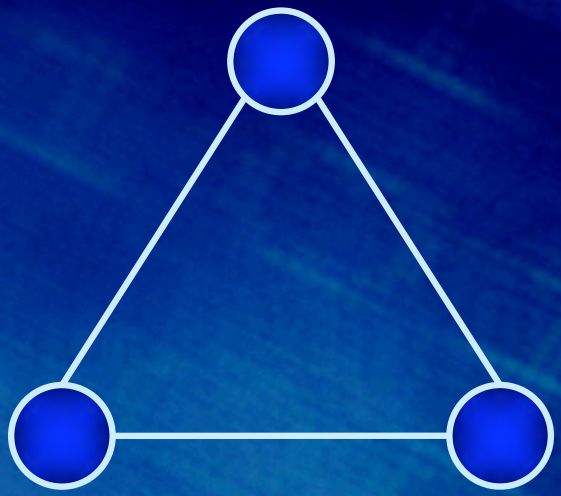
Grafi completi

Se l'insieme E di un grafo semplice contiene tutte le possibili coppie (ordinate o no a seconda del tipo di grafo) il grafo si dice completo.

Il grafo completo con $|V|$ nodi viene solitamente indicato con $K_{|V|}$

Rel. 7.04.2019

Grafi completi: Esempi



K_3

K_4

K_5

Rel. 7.04.2019

Dimensioni di un grafo completo

$K_{|V|}$ ha esattamente:

- **$|V| \cdot (|V| - 1)$ spigoli nel caso di grafi orientati**
- **$|V| \cdot (|V| - 1) / 2$ spigoli nel caso di grafi non orientati.**

Grado di un nodo

- In un grafo, orientato o no, il **grado o valenza** di un nodo *v* è il numero $\rho(v)$ di archi incidenti il nodo stesso.
- In un grafo orientato il **grado in ingresso** $\rho^{\rightarrow}(v)$ di un vertice *v* è il numero di archi incidenti sul vertice stesso.
- In un grafo orientato il **grado in uscita** $\rho^{\leftarrow}(v)$ di un vertice *v* è il numero di archi incidenti dal vertice stesso.

Rel. 7.04.2019

Teorema

- **La somma dei gradi dei vertici di un grafo, orientato o no, è un numero pari, a condizione che si assuma che un cappio dia un contributo pari a 2 nella valutazione del grado del vertice cui è incidente.**
- **Tale somma vale 2|E|.**

Corollario

- **In un qualsiasi grafo, esiste un numero pari di vertici aventi ordine dispari.**

Grafi regolari

- Un grafo in cui tutti i nodi hanno lo stesso grado *r* viene detto **regolare di grado *r***.
- I grafi regolari di grado 3 rivestono particolare importanza nei problemi di colorazione e vengono detti **cubici o trivalenti**.
- Un grafo semplice completo $K_{|V|}$ è regolare di grado $|V| - 1$.
- Un grafo completo $K_{|V|}$ è regolare di grado $|V|$.

Rel. 7.04.2019

Grafi pesati

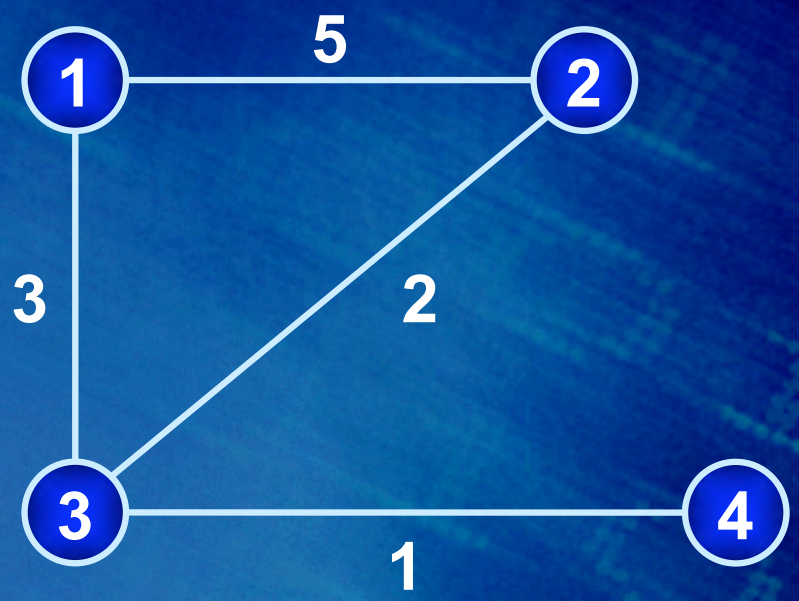
Un grafo **pesato** è una tripla ordinata $G = (V, E, P)$ tale che:

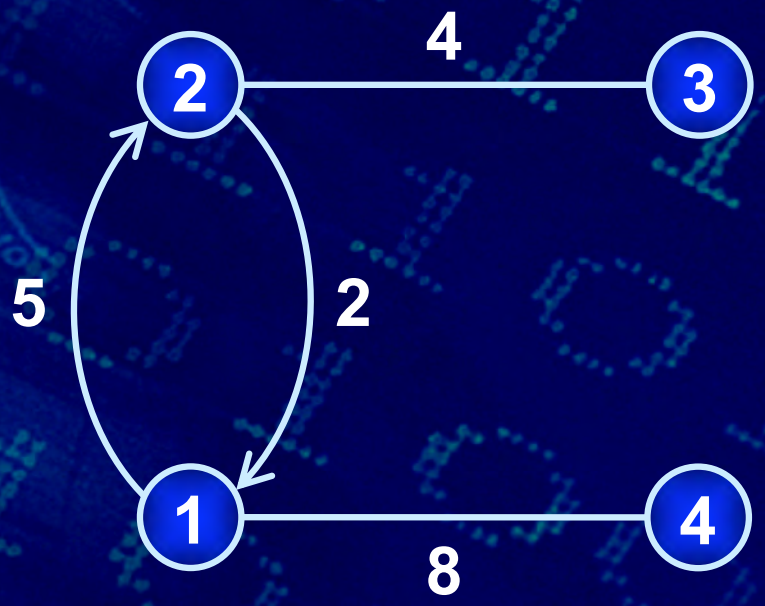
- $G' = (V, E)$ è un grafo
- P è una funzione che associa ad ogni arco (v_i, v_j) o $\{v_i, v_j\}$ di G' un peso $p_{i,j} \in \mathcal{R} \cup \{+\infty, -\infty\}$.

Il modo usuale di rappresentare i grafi pesati consiste nell'indicare, in vicinanza di ogni arco, il peso corrispondente.

Rel. 7.04.2019

Grafi pesati: Esempi





Rel. 7.04.2019

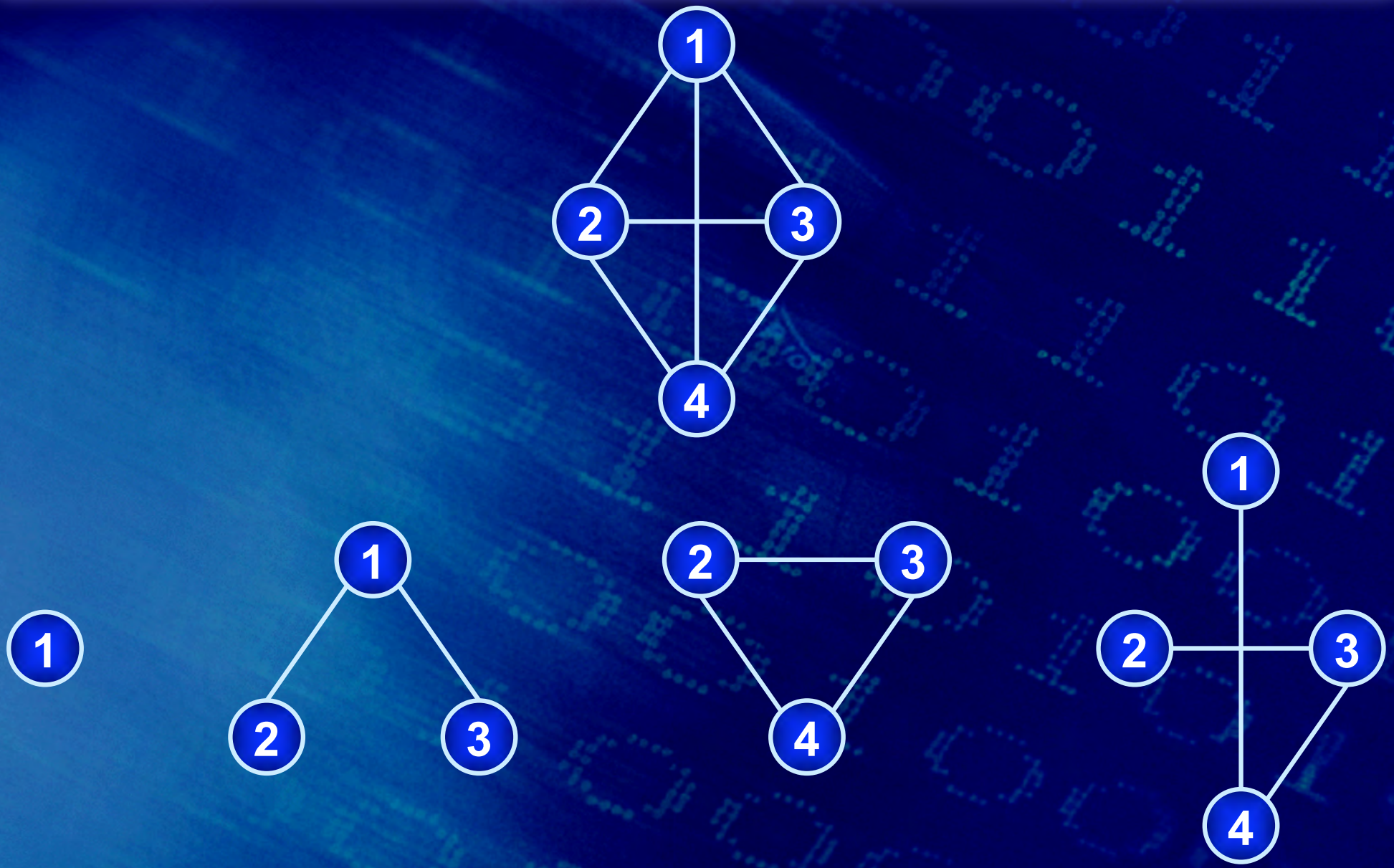
Sottografo

Dato un grafo $G = (V, E)$ si dice sottografo di G un grafo G' tale che, se $G' = (V', E')$, si abbia:

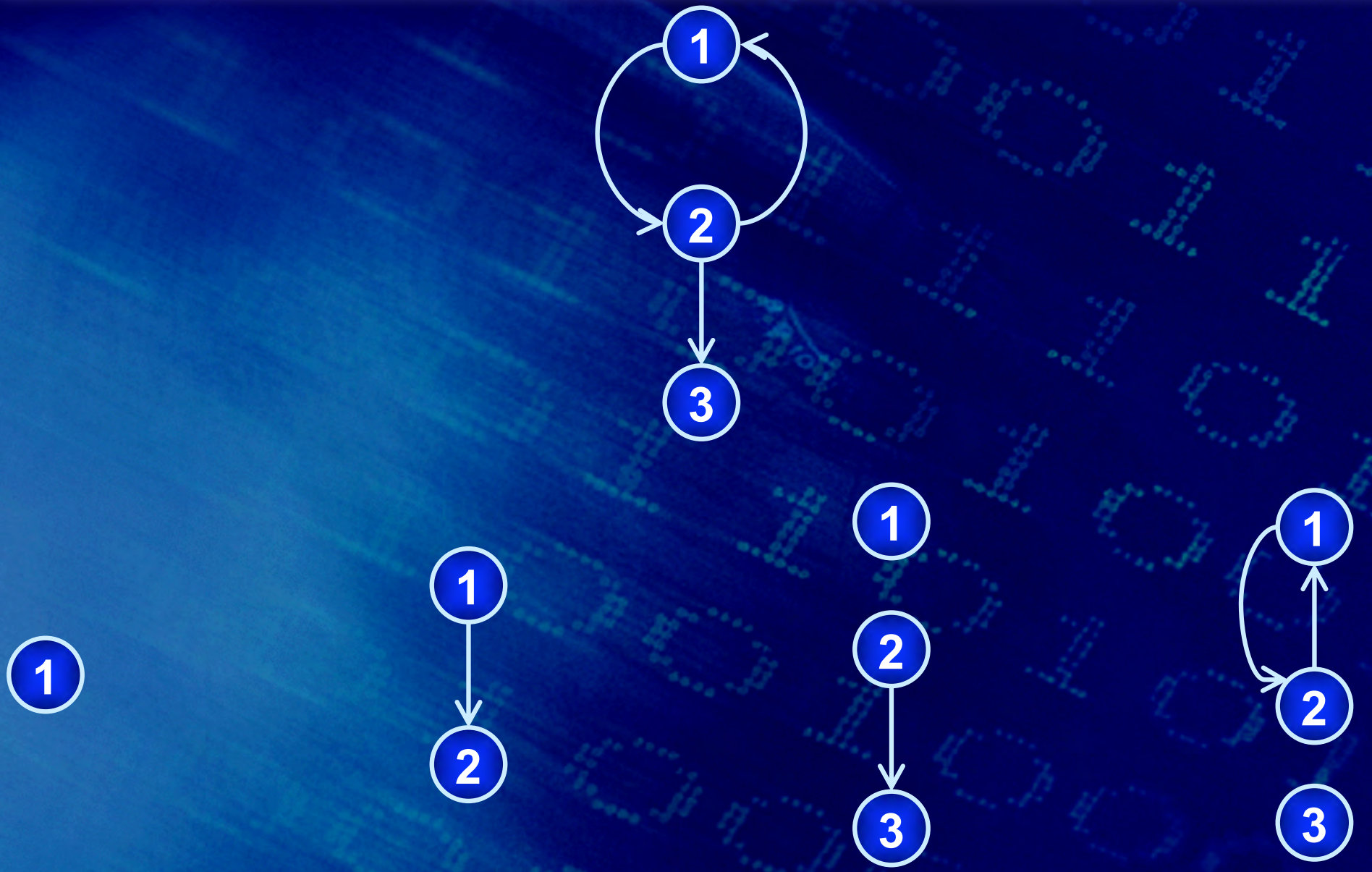
- $V' \subseteq V$
- $E' \subseteq E$

Rel. 7.04.2019

Sottografo: Esempi di grafi non orientati



Sottografo: Esempi di grafi orientati



Grafi bipartiti

- **Un grafo *G* si dice bipartito se è possibile suddividere l'insieme *V* dei suoi nodi in due sottoinsiemi disgiunti V_1 e V_2 in modo tale che ogni spigolo di *G* congiunga un nodo di V_1 a un nodo di V_2.**

Grafi bipartiti completi

- Un grafo non orientato bipartito in cui ogni nodo di V_1 sia congiunto a ogni nodo di V_2, viene definito **bipartito non orientato completo** e indicato con

  $$K_{m,n}$$

  essendo m ed n rispettivamente il numero di nodi di V_1 e V_2.
- $K_{m,n}$ ha $m + n$ vertici e $m \cdot n$ spigoli.

Rel. 7.04.2019

Grafi bipartiti completi

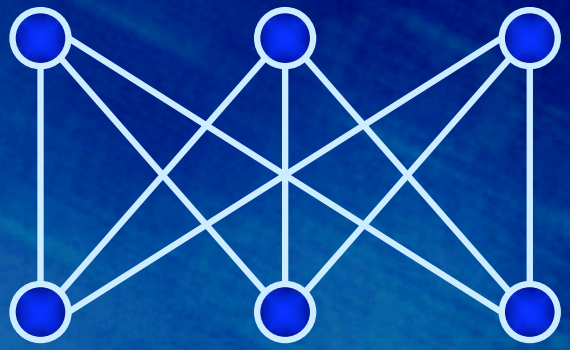
- Un grafo orientato bipartito in cui ogni nodo di V_1 sia congiunto a ogni nodo di V_2 e ogni nodo di V_2 sia congiunto a ogni nodo di V_1, viene definito **bipartito orientato completo** e indicato con

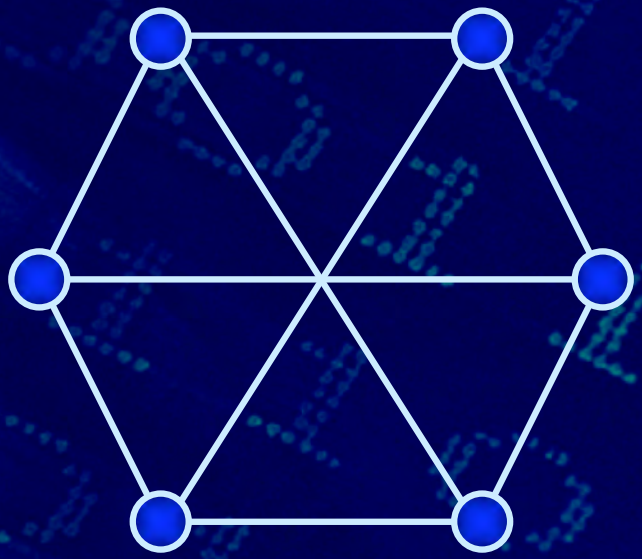
  $$K_{m,n}$$

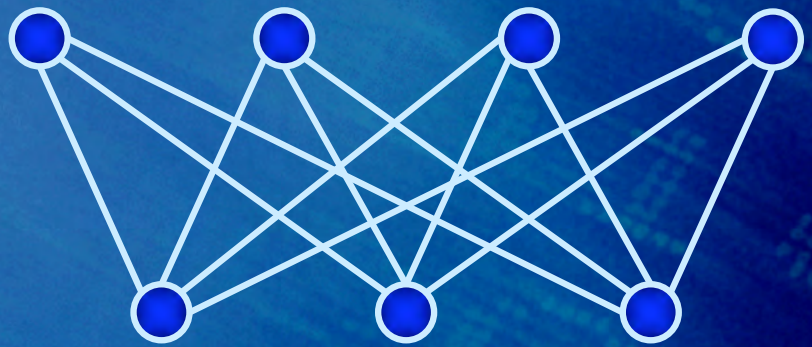
  essendo m ed n rispettivamente il numero di nodi di V_1 e V_2.
- $K_{m,n}$ ha $m + n$ vertici e $2 \cdot m \cdot n$ spigoli.

Rel. 7.04.2019

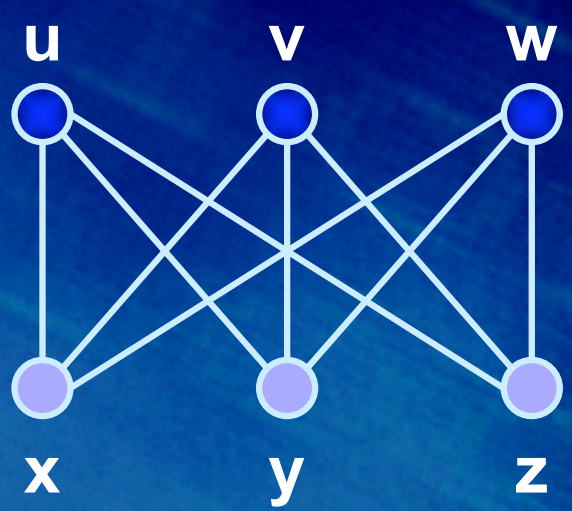
Grafi bipartiti completi: Esempi

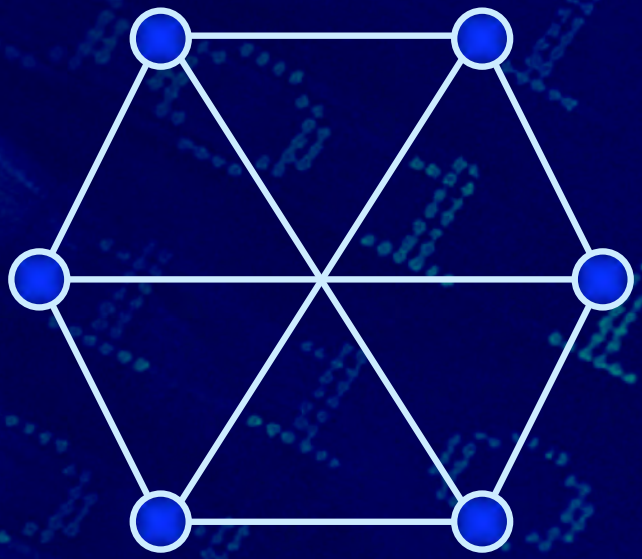


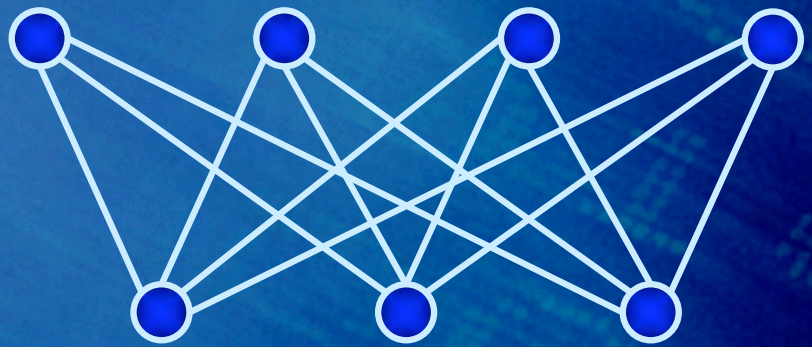




Grafi bipartiti completi: Esempi

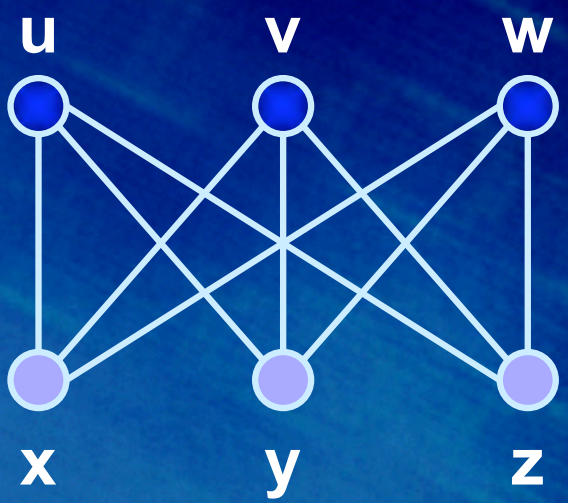


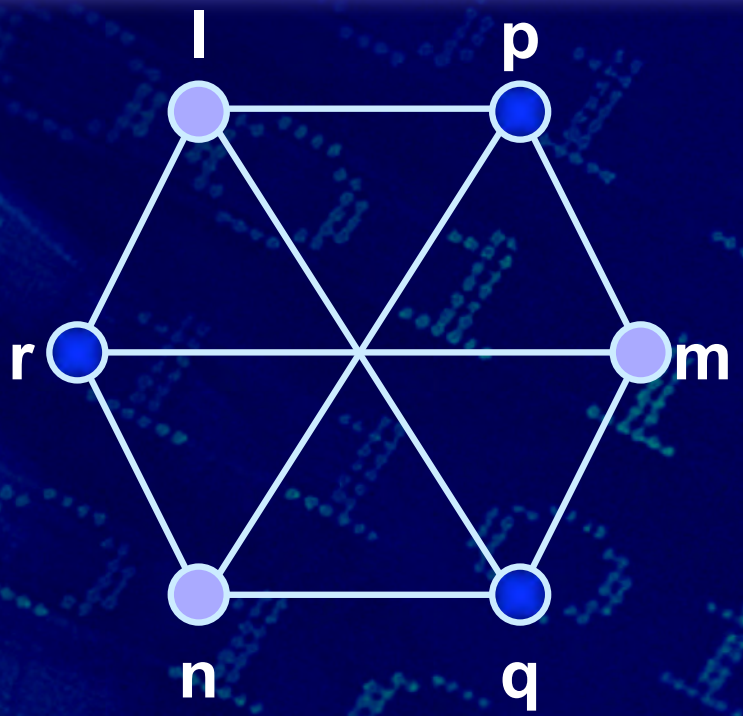


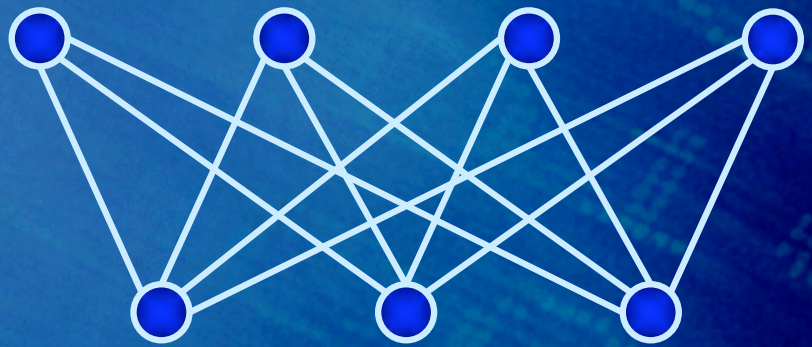


Rel. 7.04.2019

Grafi bipartiti completi: Esempi

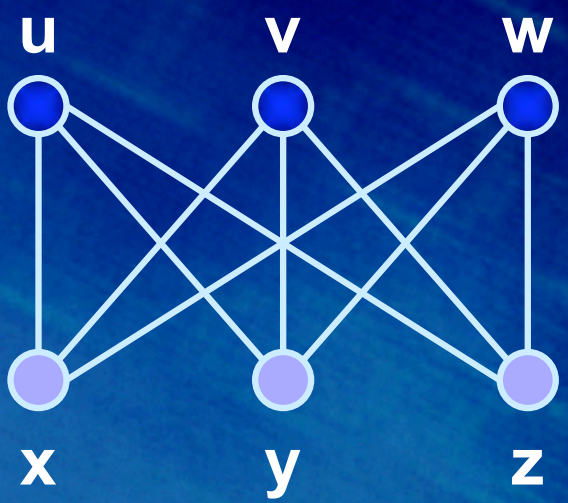


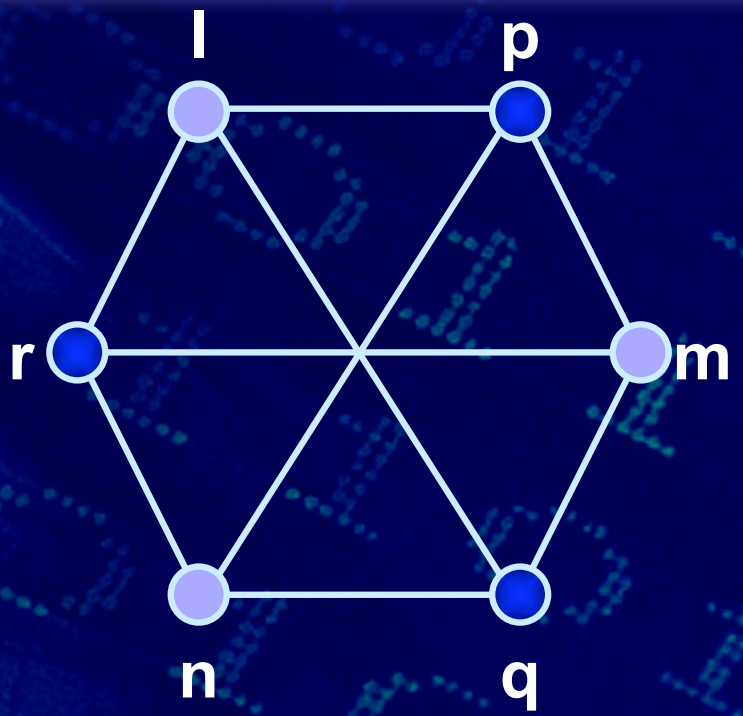


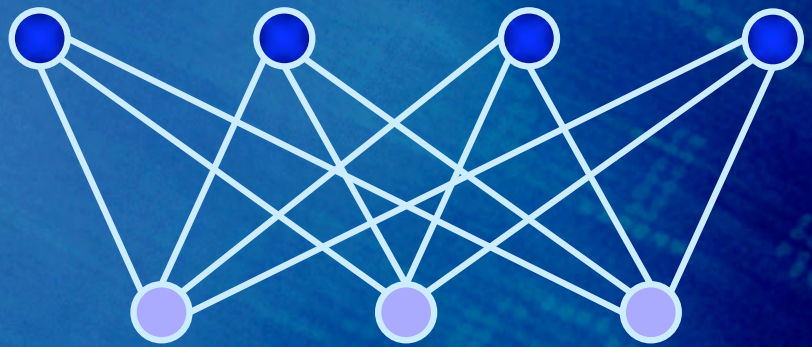


Rel. 7.04.2019

Grafi bipartiti completi: Esempi

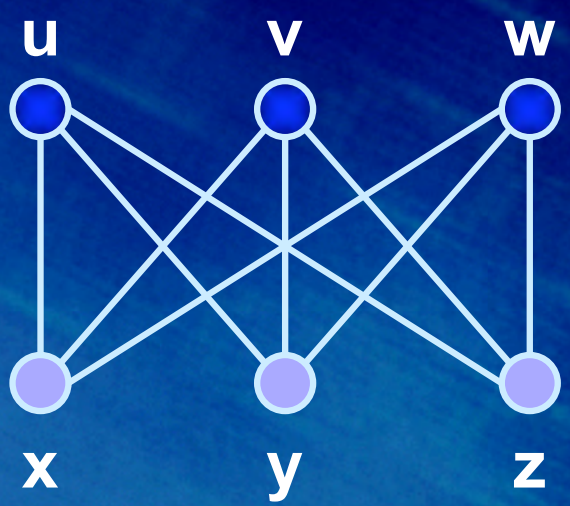


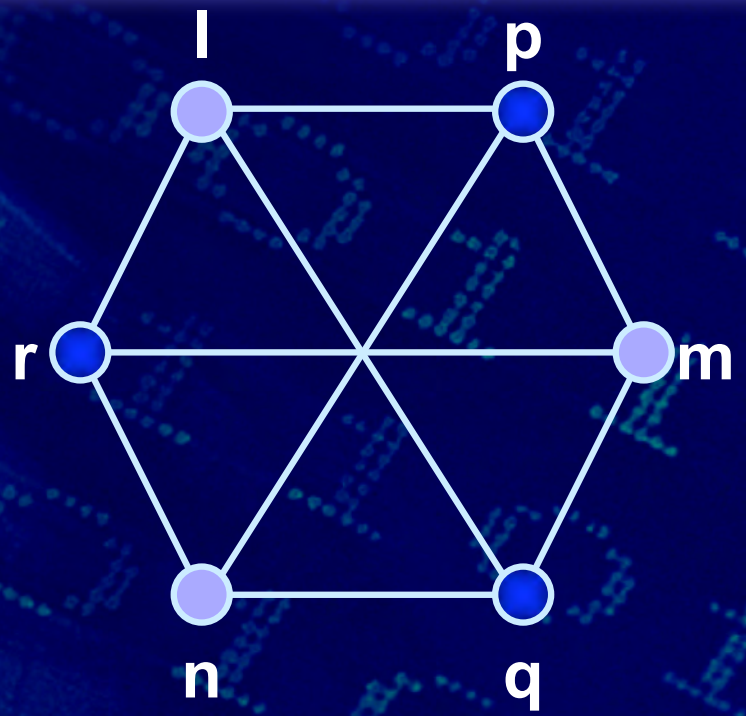


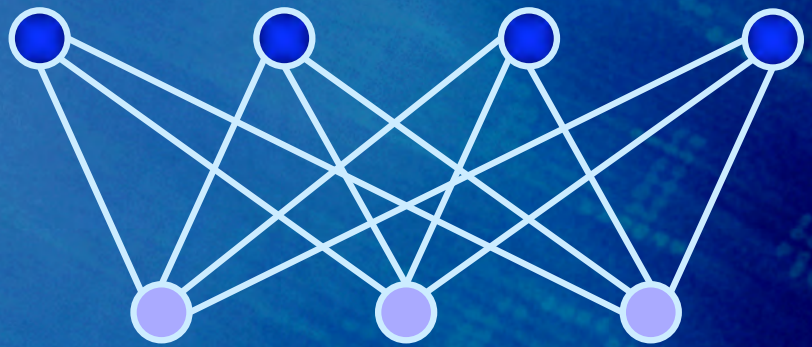


Rel. 7.04.2019

Grafi bipartiti completi: Esempi







Rel. 7.04.2019

Outline

- Introduzione
- Definizioni varie
- Rappresentazione di un grafo

Rel. 7.04.2019

Rappresentazione di un grafo

- **La rappresentazione dei grafi all'interno dei sistemi di elaborazione viene solitamente effettuata ricorrendo a:**
  - ***matrice di adiacenza***
  - ***liste di adiacenza***
  - ***matrice di incidenza***

Rappresentazione di un grafo

- **La rappresentazione dei grafi all'interno dei sistemi di elaborazione viene solitamente effettuata ricorrendo a:**
  - ***matrice di adiacenza***
  - ***liste di adiacenza***
  - ***matrice di incidenza***

Matrice di adiacenza

La **matrice di adiacenza** o **matrice di connessione** di un grafo non pesato è una matrice A di dimensioni $|V| \times |V|$, il cui generico elemento a_{ij} (per $1 \leq i,j \leq |V|$) vale:

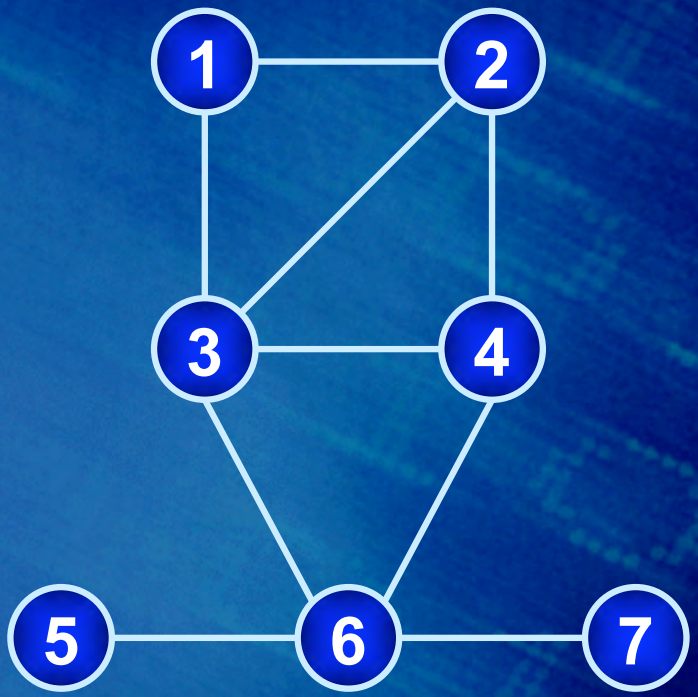
- 1 se (v_i, v_j) o $\{v_i, v_j\} \in E$
- 0 altrimenti.

Per i grafi non orientati la matrice di adiacenza è simmetrica, in quanto $a_{ij} = a_{ji}$.

Se il grafo è semplice, la diagonale principale contiene solo degli 0.

Matrice di adiacenza: esempio

Grafo non pesato e relativa matrice di adiacenza



$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Matrice di adiacenza

La **matrice di adiacenza** o **matrice di connessione** di un grafo pesato è una matrice A di dimensioni $|V| \times |V|$, il cui generico elemento a_{ij} (per $1 \leq i,j \leq |V|$) vale:

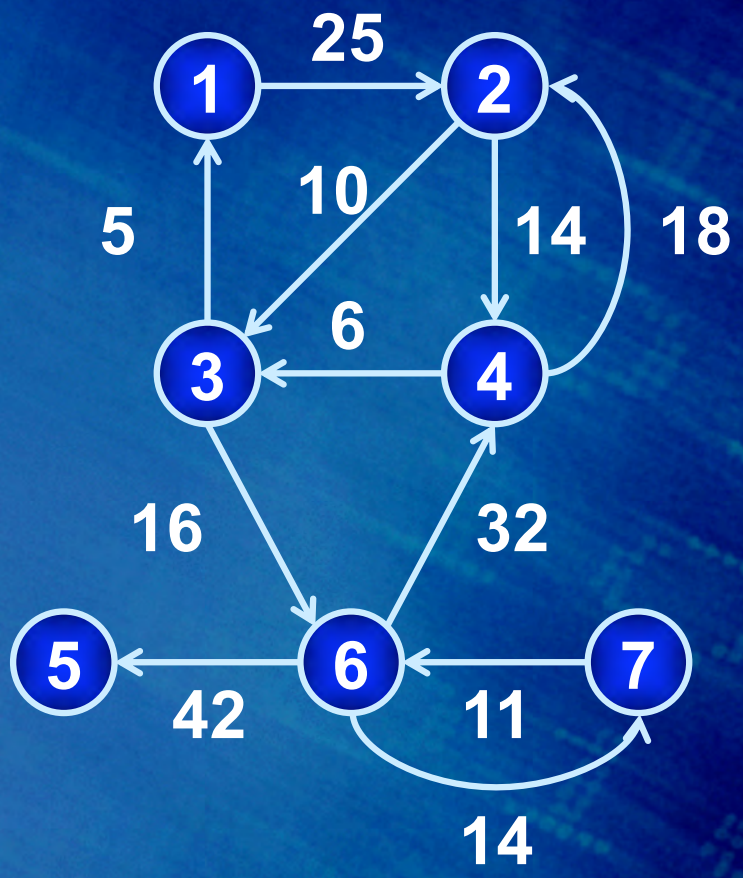
- p_{ij} se (v_i, v_j) o $\{v_i, v_j\} \in E$
- c altrimenti

dove:

- c è una costante il cui valore dipende dall'interpretazione del peso e dallo specifico problema da risolvere.

Matrice di adiacenza: esempio

Grafo pesato e relativa matrice di adiacenza



$$\begin{pmatrix} 0 & 25 & \infty & \infty & \infty & \infty & \infty \\ \infty & 0 & 10 & 14 & \infty & \infty & \infty \\ 5 & \infty & 0 & \infty & \infty & 16 & \infty \\ \infty & 6 & 18 & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 0 & \infty & \infty \\ \infty & \infty & \infty & 32 & 42 & 0 & 14 \\ \infty & \infty & \infty & \infty & \infty & 11 & 0 \end{pmatrix}$$

Rel. 7.04.2019

Matrice di adiacenza

L'occupazione di memoria risulta ovviamente $\Omega(|V|^2)$ e può essere ridotta solo ricorrendo, ove possibile, alle tecniche già viste per la memorizzazione di matrici sparse.

Rappresentazione di un grafo

- **La rappresentazione dei grafi all'interno dei sistemi di elaborazione viene solitamente effettuata ricorrendo a:**
  - ***matrice di adiacenza***
  - ***liste di adiacenza***
  - ***matrice di incidenza***

Rel. 7.04.2019

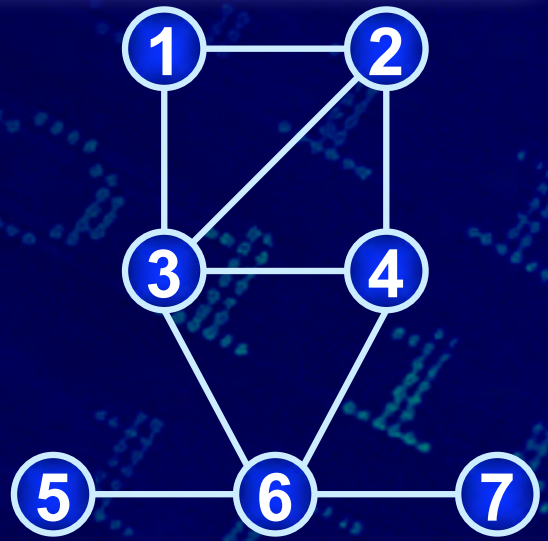
Liste di adiacenza

Nel caso in cui sia particolarmente rilevante l'informazione sugli archi esistenti, conviene ricorrere a una rappresentazione tramite liste di adiacenza:

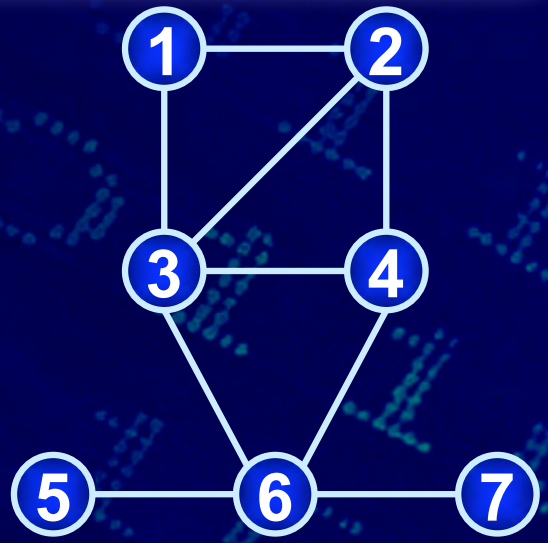
- **per i grafi non orientati: per ogni vertice si memorizza la lista dei vertici a lui adiacenti**
- **per i grafi orientati: per ogni vertice si memorizza la lista dei successori (e a volte quella dei predecessori).**

In pratica si utilizza un vettore *AL[1:|V|]* in cui *AL[i]* contiene il puntatore alla lista relativa al vertice v_i.

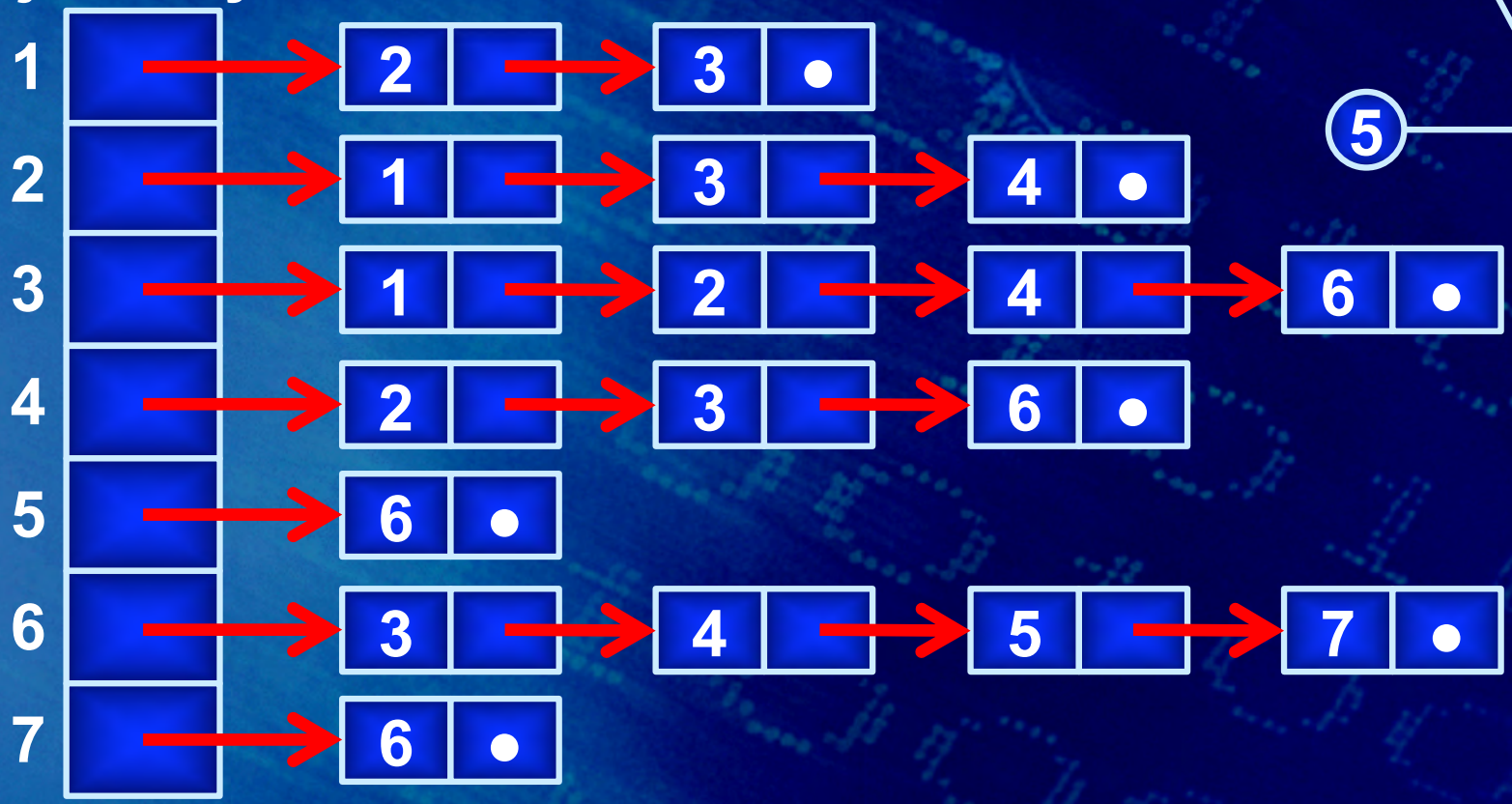
Liste di adiacenza: esempio



Liste di adiacenza: esempio

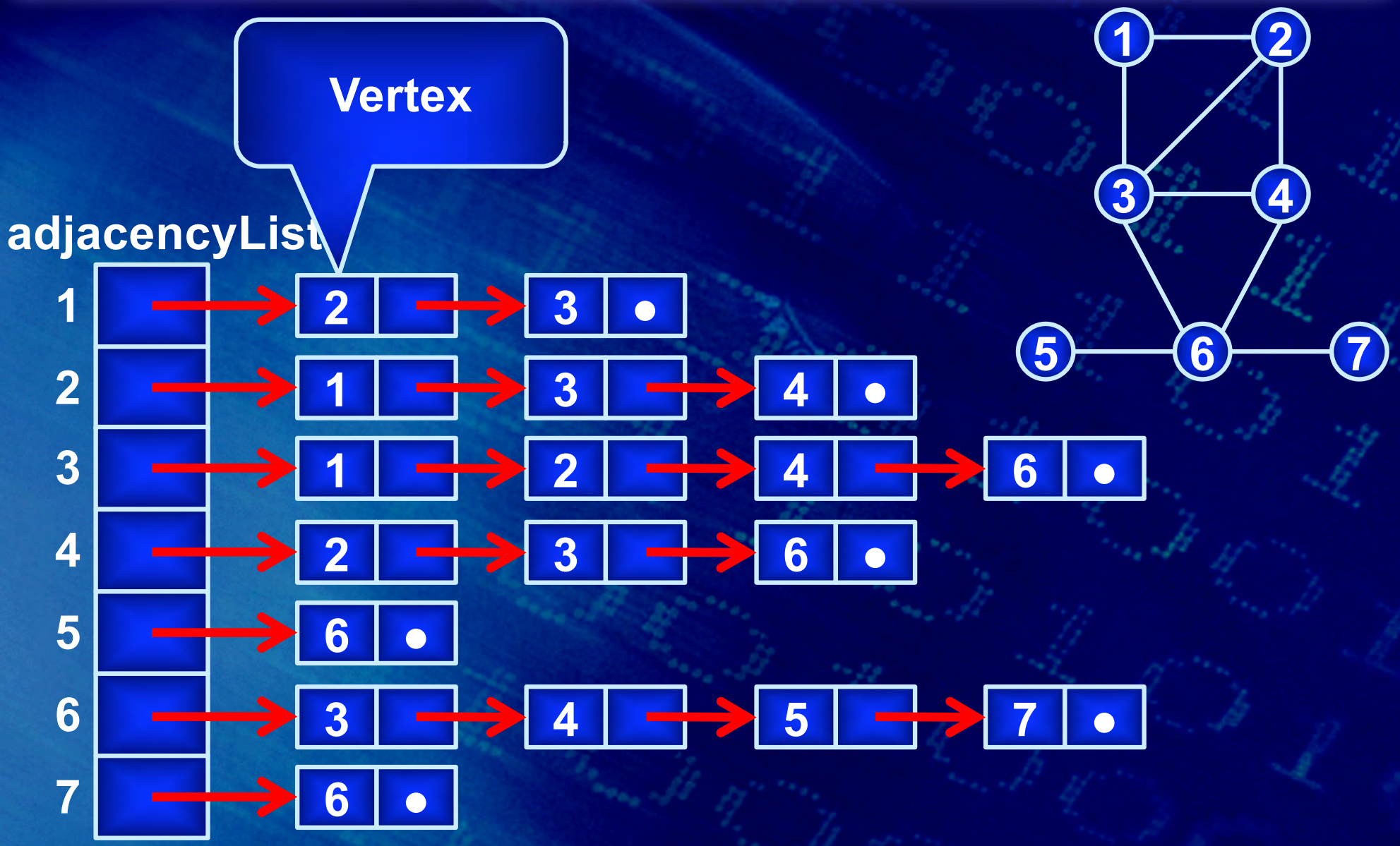


adjacencyList



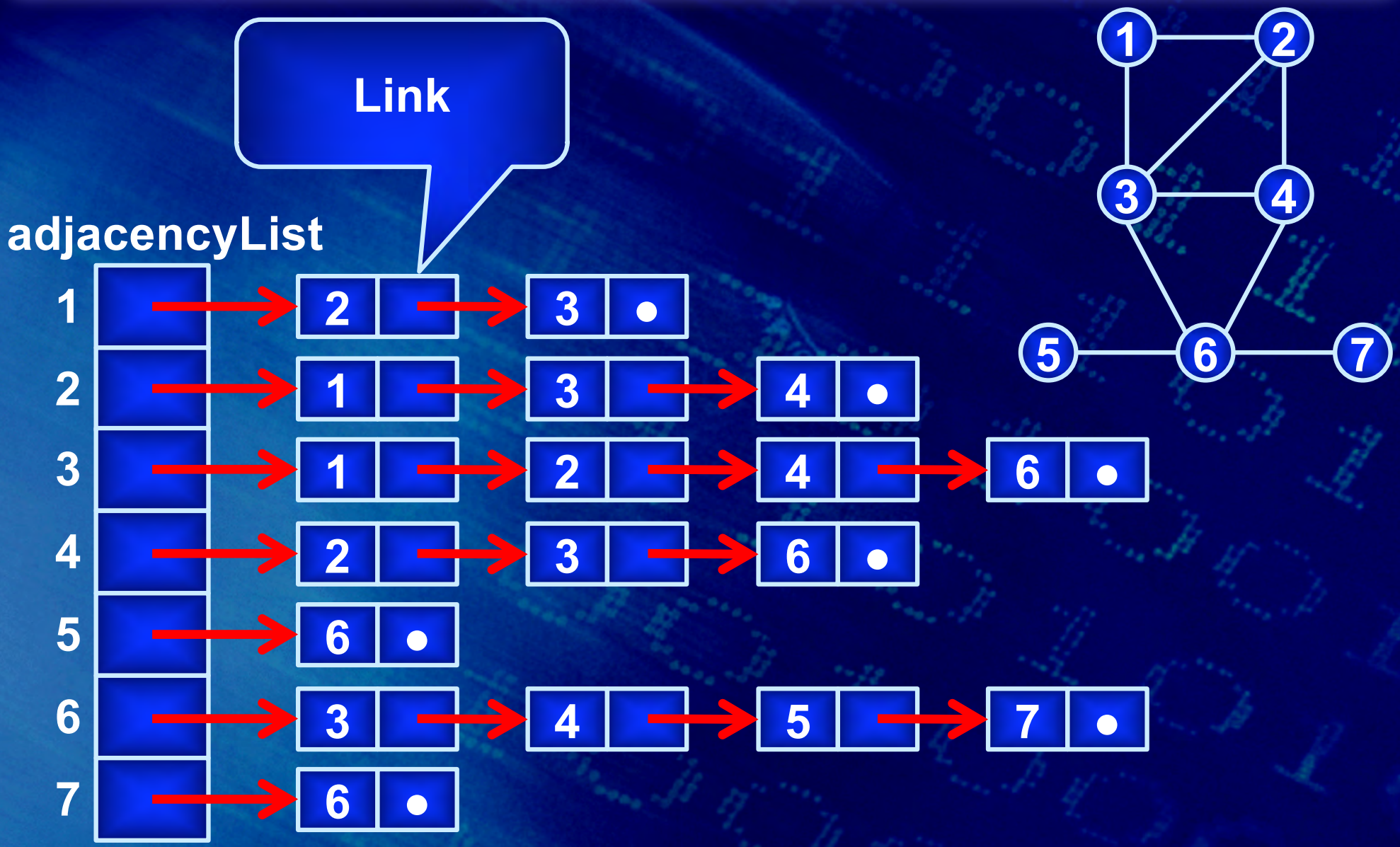
Rel. 7.04.2019

Liste di adiacenza: esempio



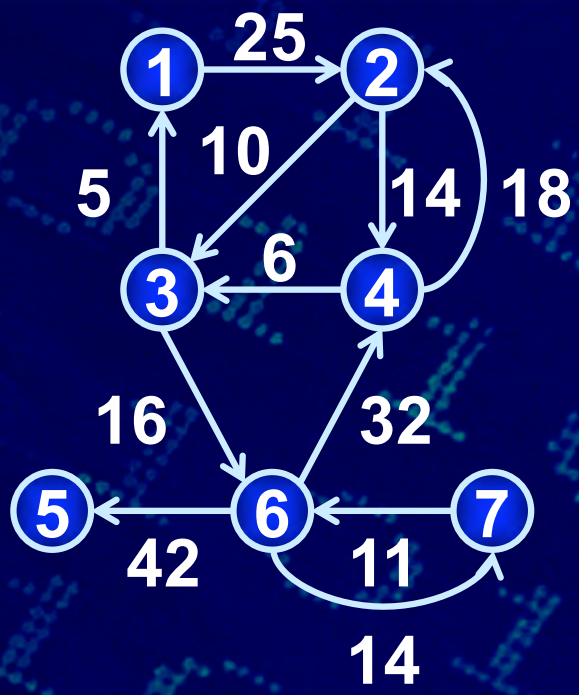
Rel. 7.04.2019

Liste di adiacenza: esempio

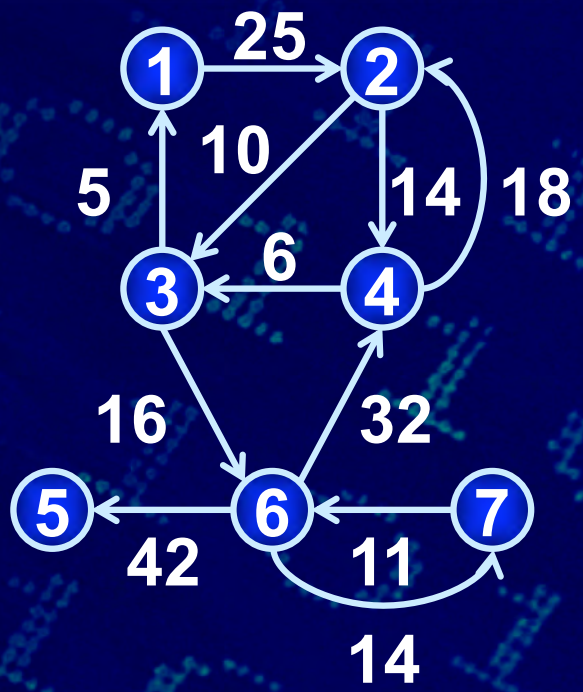


Rel. 7.04.2019

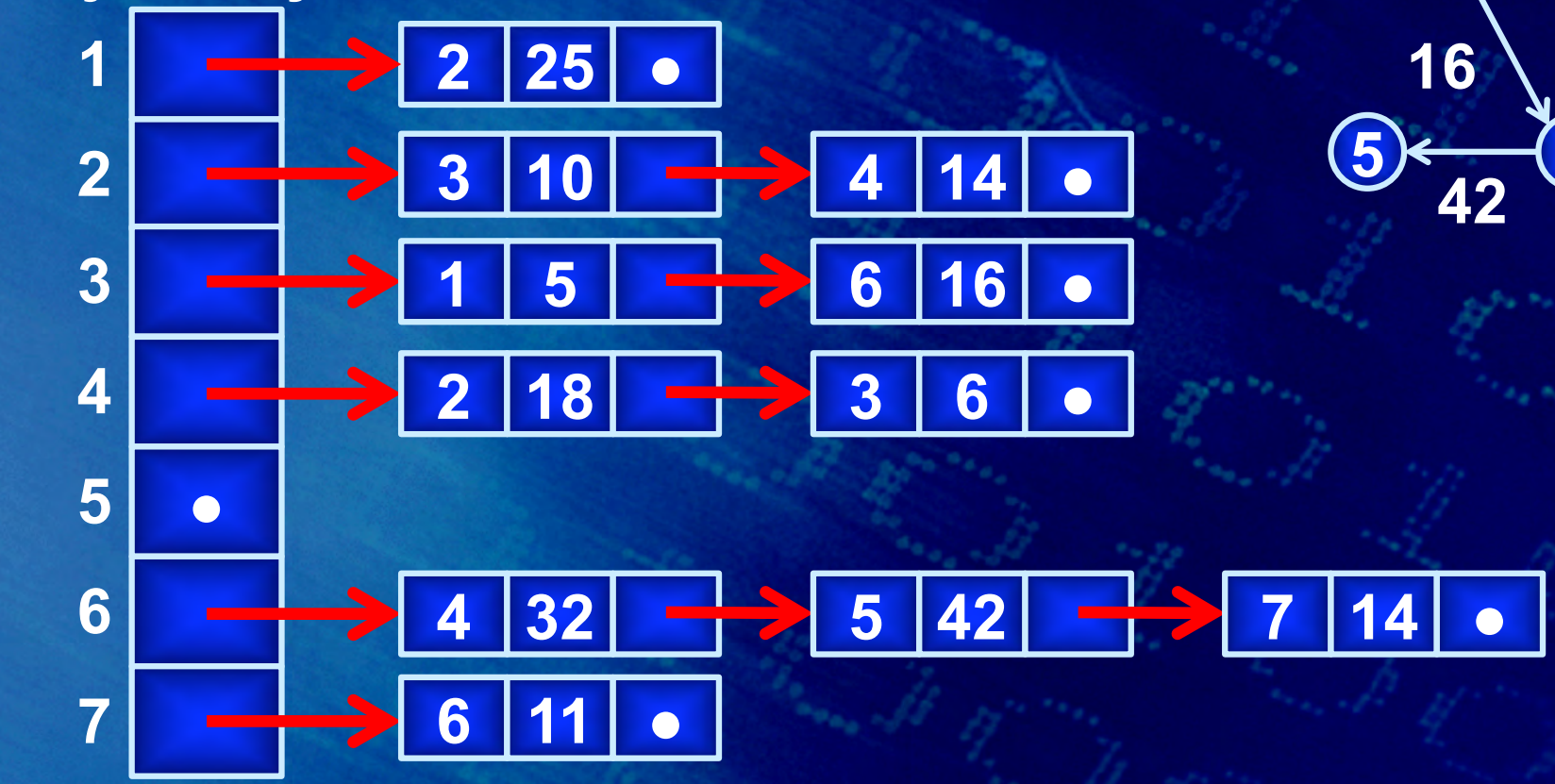
Liste di adiacenza: esempio



Liste di adiacenza: esempio

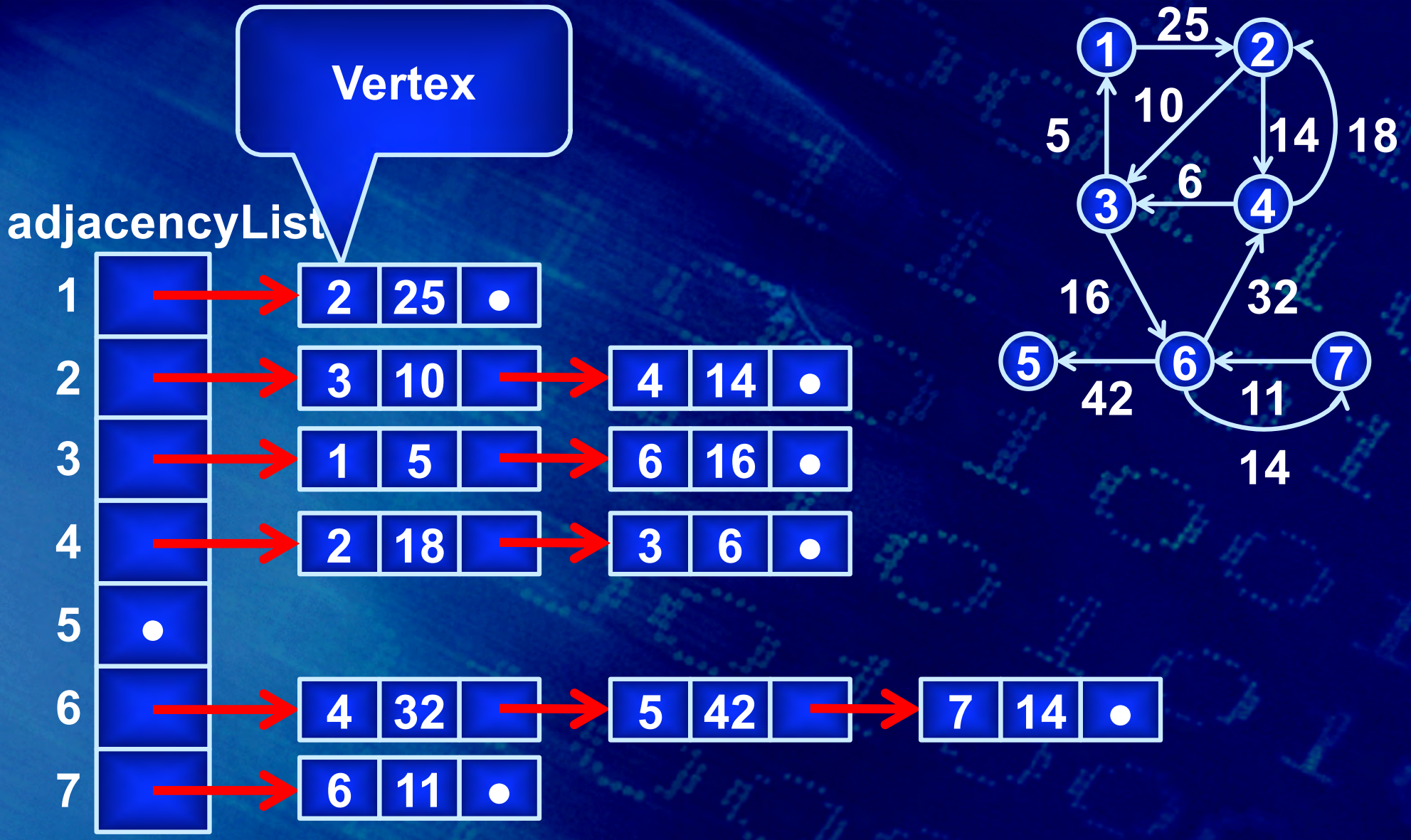


adjacencyList



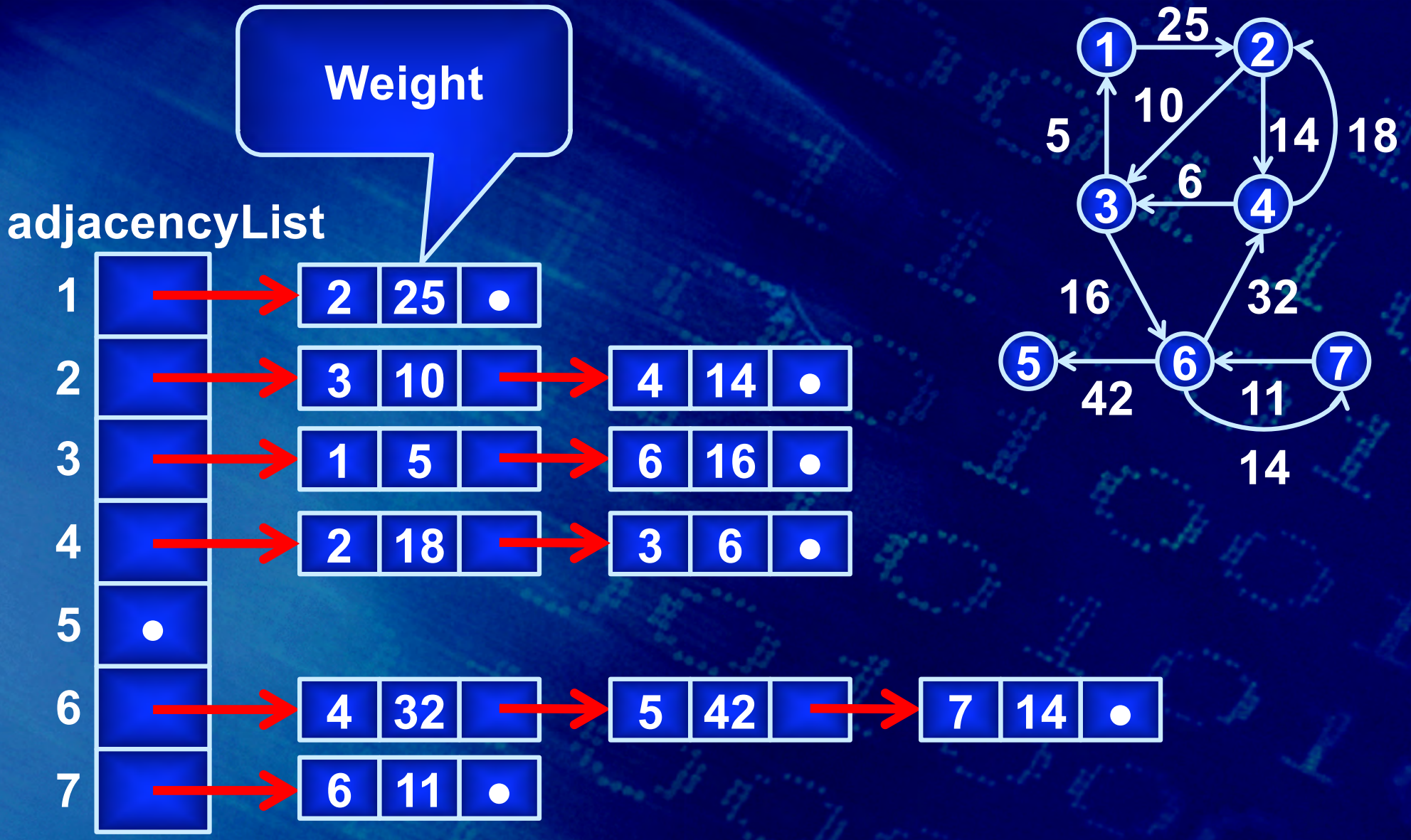
Rel. 7.04.2019

Liste di adiacenza: esempio



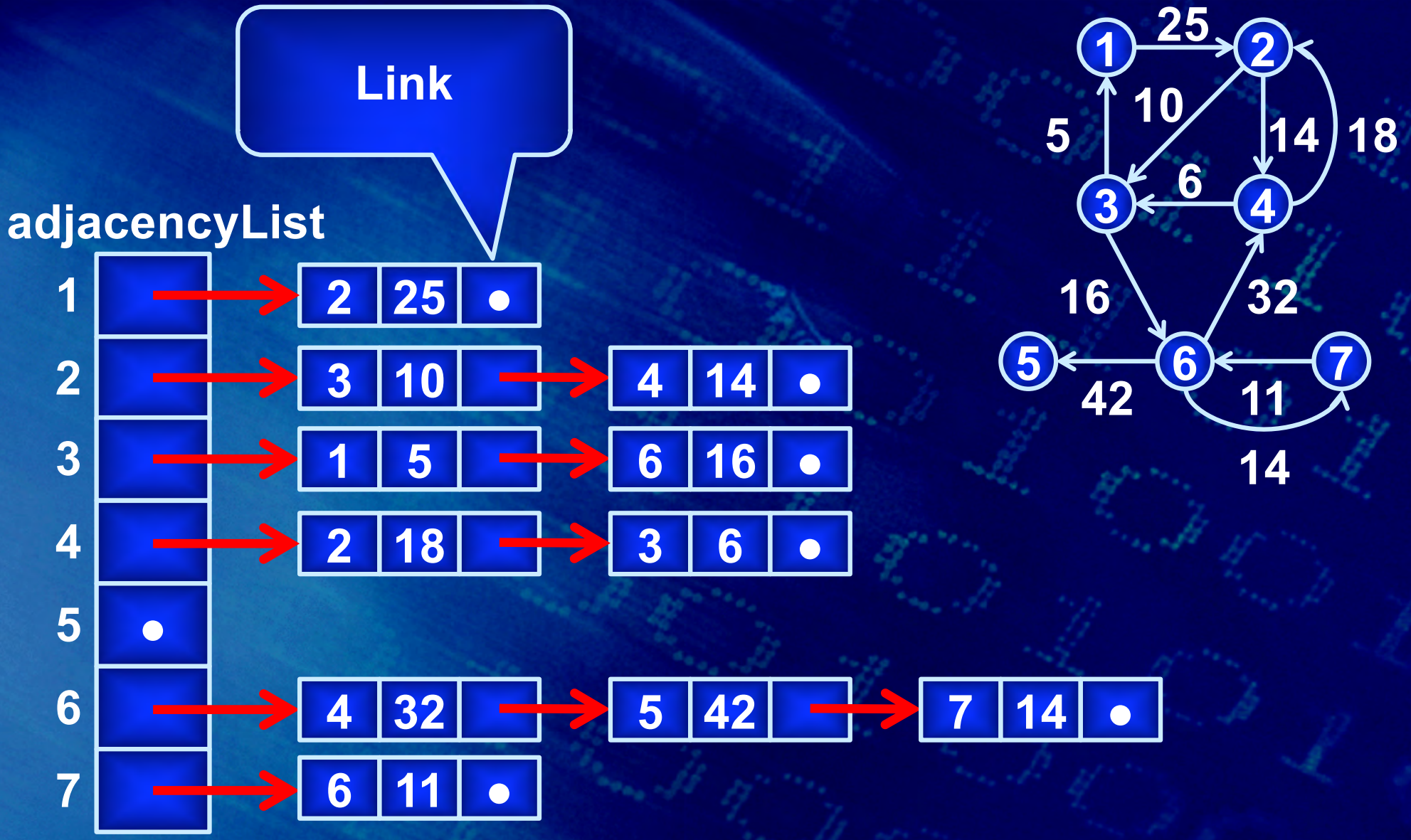
Rel. 7.04.2019

Liste di adiacenza: esempio



Rel. 7.04.2019

Liste di adiacenza: esempio



Rel. 7.04.2019

Liste di adiacenza

Le liste delle adiacenze sono meno semplici da gestire della matrice di adiacenza, ma occupano meno spazio in memoria, in quanto contengono un elemento per ogni arco esistente:

$$\Omega(|V|+|E|)$$

Risultano particolarmente adatte per grafi sparsi, in cui il numero di archi $|E|$ è molto minore di quello massimo.

Matrice di incidenza

La matrice di incidenza di un grafo non orientato è una matrice K di dimensioni $|V| \times |E|$, il cui generico elemento k_{ij} vale:

- *1* se *i* è uno dei nodi adiacenti all'arco a_j
- 0 altrimenti.

Poiché ogni arco è incidente a due vertici, in ogni colonna esistono due soli elementi pari ad *1*.

Rel. 7.04.2019

Matrice di incidenza

Nel caso dei grafi orientati, l'elemento k_{ij} diventa:

- ***+1*** **se** ***i*** **è il nodo iniziale dell'arco** a_j
- ***-1*** **se** ***i*** **è il nodo finale dell'arco** a_j
- ***0*** **altrimenti.**

Matrice di incidenza

L'occupazione di memoria è

$$\Omega(|V| \cdot |E|)$$

Ne consegue che tale rappresentazione è sempre più svantaggiosa di quella tramite liste di adiacenza e migliore di quella tramite matrice di adiacenza solo quando

$$|E| < |V|$$

cioè quando il numero di archi è minore del numero di nodi.

Per questa ragione è poco usata.

References

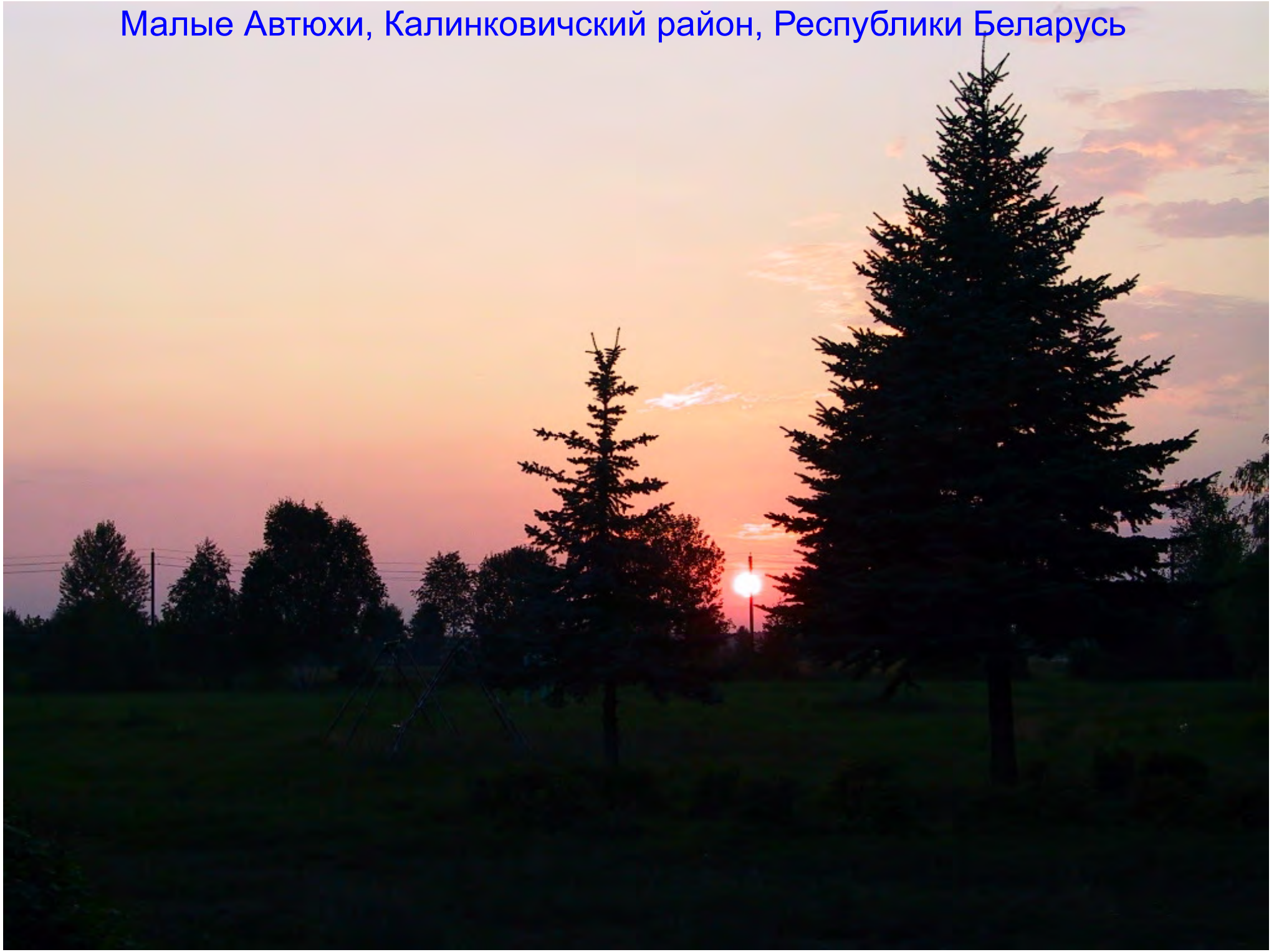
- A.V. Aho, J.E. Hopcroft, J.D. Ullman: "Data Structures and Algorithms," Addison Wesley, Reading MA (USA), 1983 pp. 198-252
- G. Ausiello, A. Marchetti-Spaccamela, M. Protasi: "Teoria e Progetto di Algoritmi Fondamentali," Ed. Franco Angeli, Milano, 1985, pp. 265-364
- E. Horowitz, S. Sahni: "Fundamentals of Computer Algorithms," Pittman, London (UK), 1978, pp. 272-325

References

- C.L. Liu:
  "Introduction to Combinatorial Mathematics,"
  McGraw-Hill Book Company, New York (USA),
  pp. 167-297
- R. Sedgewick:
  "Algorithms in C,"
  Addison Wesley, Reading MA (USA), 1990
  pp. 415-508
- C.J. Van Wyk:
  "Data Structures and C Programs," Addison
  Wesley, Reading MA (USA), 1988
  pp. 313-341

References

- **R.J. Wilson:**
  “Introduzione alla teoria dei grafi,”
  Cremonese, Roma 1978, pp. 1-155


Малые Автюхи, Калинковичский район, Республики Беларусь