

Introduzione alla Teoria dei Grafi

Algoritmi 2 – Lezione 2

A. Monti
Sapienza Università di Roma

Corso: Algoritmi 2

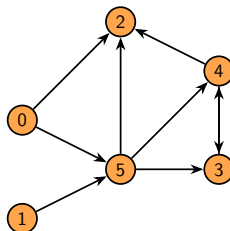
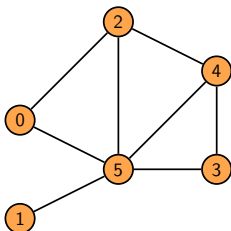
- 1 Introduzione ai Grafi
- 2 Rappresentazione dei Grafi
 - Matrice di Adiacenza
 - Lista di Adiacenza
 - Dizionario di Adiacenza
- 3 Esercizio: Pozzo Universale

Definizione di Grafo

- Un grafo è una struttura matematica che modella relazioni tra entità.
- Formalmente, un grafo è definito come una coppia $G = (V, E)$, dove:
 - V è un insieme di **nodi** (o vertici).
 - E è un insieme di **archi**, che rappresentano le relazioni tra i nodi.

Grafi Orientati vs Non Orientati

- **Grafo non orientato:** ogni arco $\{u, v\}$ rappresenta una connessione bidirezionale.
- **Grafo orientato (o digrafo):** ogni arco (u, v) è una coppia ordinata e rappresenta una connessione unidirezionale da u a v .



Grado dei Nodi e Parametri del Grafo

- **Grado di un nodo** ($degree(u)$): il numero di archi che incidono su di esso (solo per grafi non orientati).
- Per i grafi orientati, distinguiamo:
 - **Grado entrante** ($in_degree(u)$): numero di archi che arrivano al nodo.
 - **Grado uscente** ($out_degree(u)$): numero di archi che partono dal nodo.
- Notazioni standard:
 - $n = |V|$: numero di vertici.
 - $m = |E|$: numero di archi.

In ogni grafo, orientato o non orientato, vale sempre $m = O(n^2)$. Infatti:

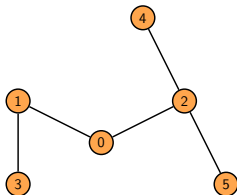
- in un grafo: $0 \leq m \leq \frac{n(n-1)}{2}$
- in un grafo orientato : $0 \leq m \leq n(n-1)$

In base al rapporto tra n ed m abbiamo:

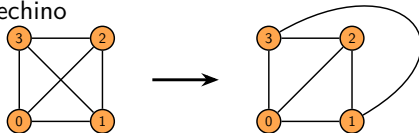
- **Grafi Sparsi:** m è proporzionale a n ovvero $m = O(n)$.
- **Grafi Densi:** m è proporzionale a n^2 ovvero $m = \Omega(n^2)$.

Alberi e Planarità

- Un **albero** è un grafo **connesso** e **aciclico**



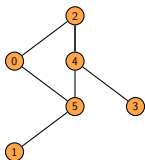
- Un **grafo planare** è un grafo che può essere disegnato sul piano senza che gli archi si intersechino



- Grafi planari ed alberi sono grafi sparsi.
 - Per un albero vale $m = n - 1$
 - Per un grafo planare con $n > 2$ vale $m \leq 3n - 6$ (Teorema di Eulero)

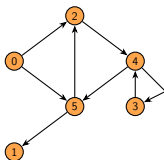
Rappresentazione tramite Matrici di Adiacenza

- Si usa una matrice binaria M di dimensione $n \times n$.
- $M[i][j] = 1$ se esiste un arco da i a j , 0 altrimenti.



Matrice di adiacenza per il grafo non orientato (sinistra)

$$M = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$



Matrice di adiacenza per il grafo orientato (destra)

$$M = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Matrici di Adiacenza - Pro e Contro

Vantaggi

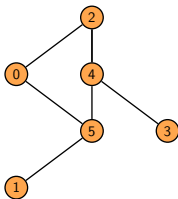
- **Verifica di un arco:** $O(1)$.
- **Implementazione:** semplice.

Svantaggi

- **Spazio:** $\Theta(n^2)$, inefficiente per grafi sparsi.
- **Iterazione sui vicini:** $\Theta(n)$.

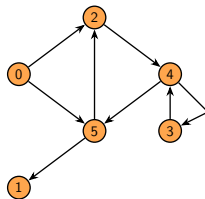
Rappresentazione tramite Liste di Adiacenza

- La lista di adiacenza è una struttura in cui ogni nodo u ha associata una lista dei nodi adiacenti.
- Lo spazio richiesto è $\Theta(n + m)$.



Lista di adiacenza per il grafo non orientato (sinistra)

```
G=[  
  [2,5],  
  [5],  
  [0,4],  
  [4],  
  [2,3,5],  
  [0,1,4]  
]
```



Lista di adiacenza per il grafo orientato (destra)

```
G=[  
  [2,5],  
  [ ],  
  [4],  
  [4],  
  [3,5],  
  [1,2]  
]
```

Vantaggi

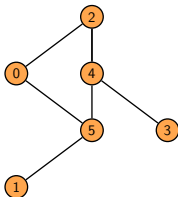
- **Spazio:** efficiente per grafi sparsi.
- **Iterazione sui vicini:** $O(\text{grado}(u))$.

Svantaggi

- **Verifica di un arco:** $O(\text{grado}(u))$, nel caso peggiore $O(n)$.
- **Implementazione:** più complessa della matrice.

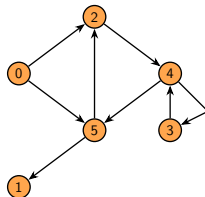
Rappresentazione tramite Dizionari

- Usati tipicamente in Python, sono simili alle liste di adiacenza.
- Ogni chiave è un nodo, il valore è una lista dei nodi adiacenti.
- Lo spazio richiesto è $\Theta(n + m)$.



Dizionario per il grafo
non orientato (sinistra)

```
G={  
0: {2, 5},  
1: {5},  
2: {0, 4},  
3: {4},  
4: {2, 3, 5},  
5: {0, 1, 4}  
}
```



Dizionario per il grafo
orientato (destra)

```
G={  
0: {2,5},  
1: { },  
2: {4},  
3: {4},  
4: {3,5},  
5: {1,2}  
}
```

Vantaggi

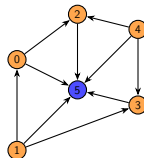
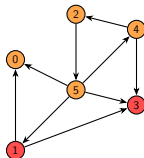
- **Spazio:** efficiente, $\Theta(n + m)$.
- **Verifica di un arco:** $O(1)$ in media.
- **Flessibilità:** nodi non numerici.

Svantaggi

- **Overhead di memoria:** leggermente superiore.
- **Caso peggiore:** la ricerca può degenerare a $O(n)$ in caso di collisioni hash.

Definizione di Pozzo Universale

- Un **pozzo** è un nodo senza archi uscenti.
- Un **pozzo universale** è un pozzo che riceve un arco da ogni altro nodo.
- Se un pozzo universale esiste, è unico.



A **sinistra** c'è un grafo con due pozzi (in rosso) mentre a **destra** c'è un grafo con un pozzo universale (in blu).

Verifica di Pozzo Universale - Idea e Matrice

$$M = \begin{matrix} & & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ & & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ & & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{matrix}$$

Un semplice algoritmo per **verificare** se un nodo x è un pozzo universale scorre la riga x per assicurarsi che contenga solo zeri (x non deve avere archi uscenti) inoltre controlla se la colonna x contiene tutti 1 tranne che nella cella $M[x][x]$ (tutti gli altri nodi devono avere un arco verso x).
L'algoritmo risolve il problema in tempo $O(n)$.

```
def test_pozzo_universale(x, M):  
    n = len(M)  
    for j in range(n):  
        if M[x][j] == 1:  
            return False  
    for i in range(n):  
        if i != x and M[i][x] == 0:  
            return False  
    return True
```

- Un semplice algoritmo esaustivo per verificare se un grafo diretto M ha un pozzo universale consiste nel controllare, per ogni nodo, se questo lo è. L'algoritmo termina appena trova un pozzo universale oppure restituisce *False*.
- Il tempo di calcolo di quest'algoritmo è $O(n^2)$, poiché per ogni nodo si eseguono fino a $2n$ controlli.

- Un algoritmo più efficiente sfrutta un'osservazione chiave: in un test $M[i][j]$, se il risultato è 1, i non può essere un pozzo universale; se è 0, non può esserlo j .
- Si escludono candidati a coppie fino a che ne rimane uno solo, che viene poi verificato.

```
def trova_pozzo_universale2(M):  
    n = len(M)  
    L = list(range(n))  
    while len(L) > 1:  
        a = L.pop()  
        b = L.pop()  
        if M[a][b] == 1:  
            L.append(b)  
        else:  
            L.append(a)  
    if not L:  
        return None  
    x = L.pop()  
    return test_pozzo_universale(x, M)
```

ESERCIZI

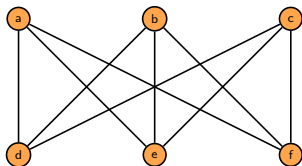
Suggerimento: Prima di cercare le soluzioni in rete, prova a ragionarci da solo: gli esercizi che seguono sono pensati per aiutarti a mettere alla prova la tua comprensione, la tua capacità di sviluppare una prova di correttezza e costruire un algoritmo corretto.

Solo dopo averci riflettuto, confronta la tua soluzione con altre possibili versioni.

Nel seguito con G si intende sempre un grafo (non diretto) con n nodi ed m archi:

- 1 Dimostrare che la somma dei gradi dei nodi di G è sempre un numero pari.
- 2 Dimostrare che in G il numero di nodi di grado dispari è sempre pari.
- 3 Dimostrare che se tutti i nodi di G hanno grado almeno due, allora nel grafo c'è almeno un ciclo. Si può affermare che se il grado di ogni vertice è esattamente due, allora G è un ciclo?
- 4 Dimostrare che se G ha almeno due nodi, allora in G ci sono due nodi con lo stesso grado.
- 5 Assumete che G sia un grafo completo di n nodi con $n \geq 3$ e che i suoi archi siano stati colorati con due colori. Qual è il valore minimo di n per cui si può essere sicuri che G contiene un ciclo di 3 nodi monocromatico?

- 6 Dimostrare che se $n = 1001$ e tutti i nodi hanno grado almeno 500, allora per ogni coppia di nodi esiste un cammino di lunghezza al più 2.
- 7 Dimostrare che il grafo di 6 nodi nella figura in basso non è planare.



- 8 Dato un grafo G il **grafo complementare** G^c è un grafo che ha gli stessi nodi di G e per archi tutte le coppie che non sono connesse in G
 - 1 Dimostrare che se $n \geq 6$, allora G o il suo grafo complementare G^c contiene un triangolo. Esibire poi un esempio di grafo G di 5 nodi tale che G e il suo grafo complementare G^c non contengono triangoli.
 - 2 Dimostrare che G o il suo grafo complementare G^c è connesso.
 - 3 Dimostrare che, se $n \geq 11$, allora G o il suo grafo complementare G^c non è planare.