NUMERO DI PERCORSI SEMPLICI

In un grafo, il numero MASSIMO di percorsi semplici tra due nodi un generico grafo e' $\sum_{i=2}^{n}(i-2)!$

Quindi questo numero cresce esponenzialmente, il che risulta essere un ostacolo per la visita.

OSS

Un grafo in cui ogni nodo e' connesso agli altri e' detto grafo COMPLETO

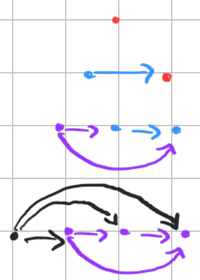
PROPOSIZIONE

Nei grafi aciclici il numero di percorsi semplici e' esponenziale ma cresce piu' lentamente che nei percorsi ciclici

DIMOSTRAZIONE

Proviamo a costruire delle classi di grafi

n	$\#\pi$		
1	1	G_1	
2	1	G_2	
3	2	G_3	
4	4	G_4	$\rightarrow \#(G_4) = \#(G_3) + \#(G_2) + \#(G_1)$



Possiamo definire la seguente equazione di ricorrenza

$$f(n) = \begin{cases} \sum_{i=1}^{n-1} f(i) & n > 1 \\ 1 & n = 1 \end{cases}$$

f indica il numero di percorsi semplici

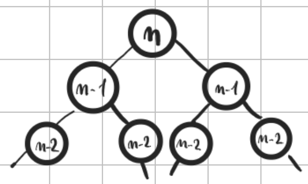
$$\sum_{i=1}^{n-1} f(i) = f(n-1) + \sum_{i=1}^{n-2} f(i) = 2f(n-1)$$

quindi l'equazione è

$$f(n) = f(n-1) + f(n-1) = 2f(n-1)$$

$$f(n) = \begin{cases} 2f(n-1) & n > 1 \\ 1 & n = 1 \end{cases}$$

Scriviamo l'albero di ricorrenza



L'albero di ricorrenza è un albero binario pieno, quindi ha 2^h nodi, ma $h = n$ quindi 2^n nodi ovvero 2^n percorsi semplici

PROPOSIZIONE

Sia $G = (V, E)$ un grafo e $s, u \in V$, u è RAGGIUNGIBILE da s se

$$\exists \pi \in PATH(G) \mid \pi_1 = s \wedge \pi_j = u \quad \text{con } j \geq 1$$

Dove $PATH(G)$ è l'insieme dei percorsi di G e π_i è l'i-esimo elemento di π

OSS

l'insieme E è sottoinsieme dell'insieme delle coppie di nodi raggiungibili tra di loro (che chiameremo REACH)
In particolare REACH è il completamento riflessivo e transitivo di E (ammette la coppia (u,u) e i percorsi di raggiungibilità applicano la transitività sugli archi)

RAPPRESENTAZIONE DEI GRAFI

Esistono (almeno) due possibili rappresentazioni per i grafi:

- Matrice di adiacenza
- Liste di adiacenza (più utilizzata in alcuni ambiti)

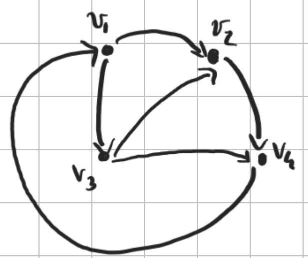
MATRICE DI ADIACENZA

E' una matrice in cui le righe e le colonne sono rappresentate dai nodi del grafo (etichette).
Dati due vertici v_i, v_j, $(v_i, v_j) \in E \iff M(v_i, v_j) = 1$

LISTA D'ADIACENZA

Dato un insieme di vertici (sottoforma di array potenzialmente), ad ogni vertice è associata una lista di nodi adiacenti (ovvero collegati da un arco singolo)

Es.



	v_1	v_2	v_3	v_4
v_1	0	1	1	0
v_2	0	0	0	0
v_3	0	1	0	1
v_4	1	0	0	0

$v_1 \to \boxed{v_3} \to \boxed{v_2}$

$v_2 \to \boxed{v_4}$

$v_3 \to \boxed{v_4} \to \boxed{v_2}$

$v_4 \to \boxed{v_1}$

OSS

- Aggiungere un nuovo nodo richiede tempo quadratico per la matrice (dobbiamo copiare la matrice) mentre lineare con la lista d'adiacenza
- le rappresentazioni non sono uniche perché non ordinate

DIMENSIONE DI UN GRAFO

Per DIMENSIONE di un grafo definiamo

$|G| = \Theta(|V| + |E|)$

VISITA SU GRAFO

Vediamo come eseguire la visita di un grafo al fine di verificare la raggiungibilità tra 2 nodi.

Scegliamo un nodo sorgente (simil-radice).
Sfruttiamo il concetto di distanza tra nodi dove gli adiacenti ad s sono a distanza 1 e così via.
Si noti che i nodi non raggiungibili da s non verranno visitati (non ci interessa per il nostro fine).
Per la ciclicità dei grafi e l'infinitezza dei percorsi non possiamo riutilizzare l'algoritmo di BFS sugli alberi perchè andrebbe in loop essendoci nodi adiacenti tra loro.
Dobbiamo verificare che ogni nodo venga visitato solo una volta