API - esercitazione 14

03/06/2021

# Grafi

$G = (V, E)$

$V \to$ nodi

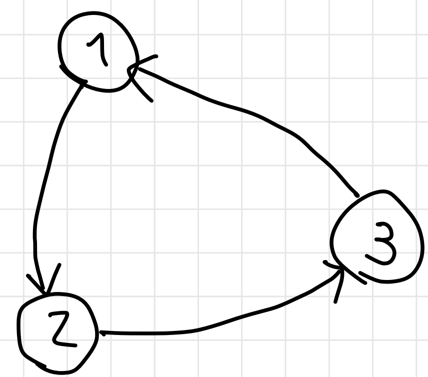
$E \to$ archi

$E \subseteq V \times V$

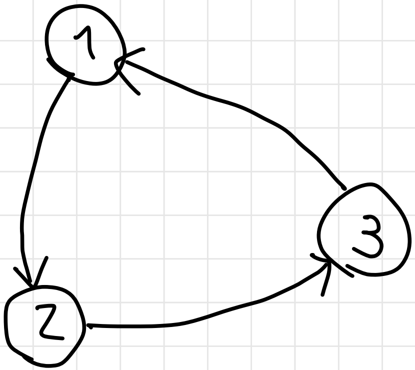
Rappr. grafica:

$V = \{1, 2, 3\}$

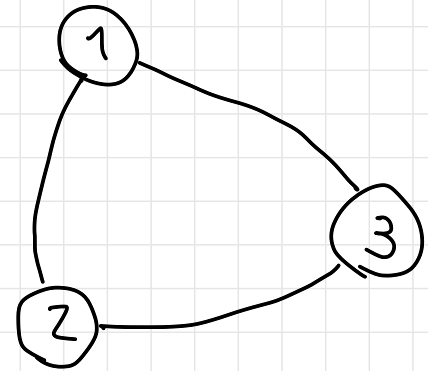
$E = \{(1,2), (2,3), (3,1)\}$



Grafo orientato:



Grafo non orientato:



$$(v_1, v_2) \in E \Rightarrow (v_2, v_1) \in E$$

Rappr. in mem.:

Liste di adiacenza:

$v_1 \; \bullet \to v_2 \to v_3$

$v_2 \; \bullet \to \text{NIL}$

$v_3 \; \bullet \to v_1$

$\vdots$

$v_{|V|} \; \bullet$

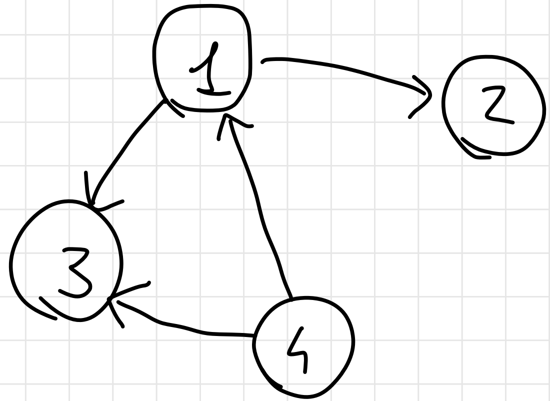
Sono i vertici adiacenti a $v_1$

$len = |V|$

Matrice di adiacenza:

$|V| \times |V|$

$$\begin{array}{c c}
\begin{array}{c} v_1 \\ v_2 \\ \vdots \end{array} &
\begin{bmatrix} \emptyset & 1 & \\ & & \\ & & \end{bmatrix} \\
& \begin{array}{ccc} v_1 & v_2 & \cdots \end{array}
\end{array}$$




Liste di ad.:

1 → 2 → 3

2

3

4 → 3 → 1

Matrice:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}
\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Complessità spaziale:

Liste : $\Theta(|V| + |E|)$

Matrice : $\Theta(|V|^2)$

$|E| \approx |V|^2$

↓

allora conviene la matrice

Complessità temporale:

Det. $(v_1, v_2) \in E$

Liste : $O(|V|)$

Matrici : $O(1)$

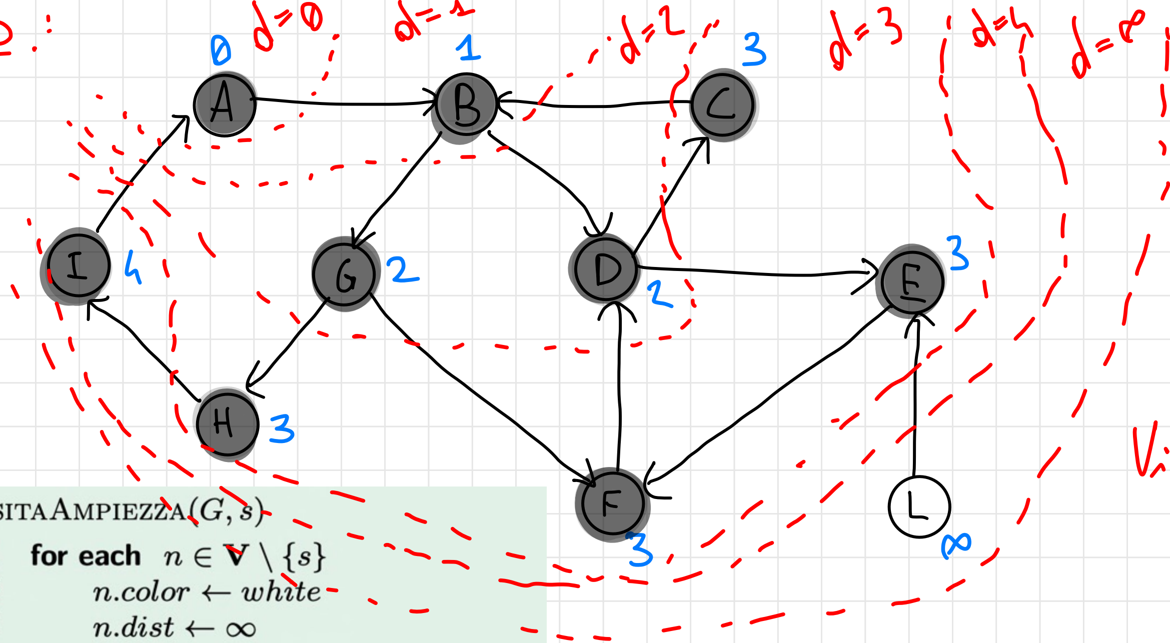
Det. num. di archi uscenti da un nodo

$o_e$

Liste : $\Theta(o_e)$

Matrici : $\Theta(|V|)$

Operazione di visita : visitare i nodi raggiungibili da una sorgente es:




Visita in ampiezza (BFS) a partire da A

```
VisitaAmpiezza(G, s)
for each n ∈ V \ {s}
    n.color ← white
    n.dist ← ∞
s.color ← grey
s.dist ← 0
Q ← ∅
Enqueue(Q, s)
while ¬IsEmpty(Q)
    curr ← Dequeue(Q)
    for each v ∈ curr.adiacenti
        if v.color = white
            v.color ← gray
            v.dist ← curr.dist + 1
            Enqueue(Q, v)
    curr.color ← black
```

curr = H

curr.adj = I

~~I~~ ~~H~~ ~~F~~ ~~E~~ ~~C~~ ~~G~~ ~~D~~ ~~B~~ ~~A~~ → Q

Complessità BFS : $O(|V|+|E|)$
DFS

Dijkstra : $O((|V|+|E|)\log|V|)$

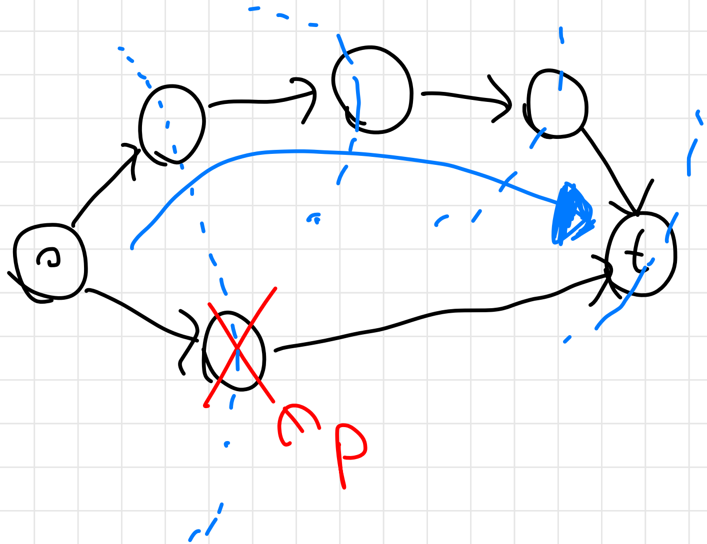
es.: Dato $G = (V, E)$, $s, t \in V$

$s$ sorgente, $t$ target

$P \subseteq V$

nodi proibiti

trovare cammino min. tra $s$ e $t$ senza nodi $\in P$



1ª idea: Dijkstra tra s e t con costo dei $n \in P = +\infty$

$$O((|V| + |E|) \log |V|)$$

2ª idea: BFS a partire da s modificato:

coloro di nero i $n \in P$

eseguo il resto di BFS normalmente

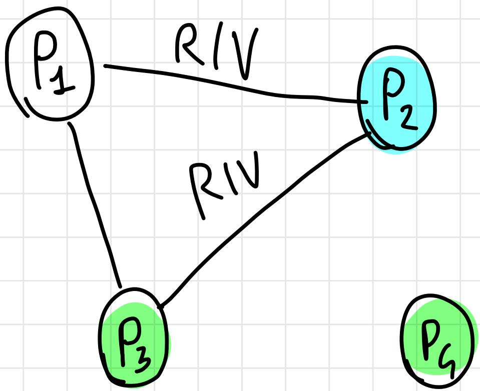
nel momento in cui trovo t ritorno

$$O(|V| + |E|) \quad \checkmark$$

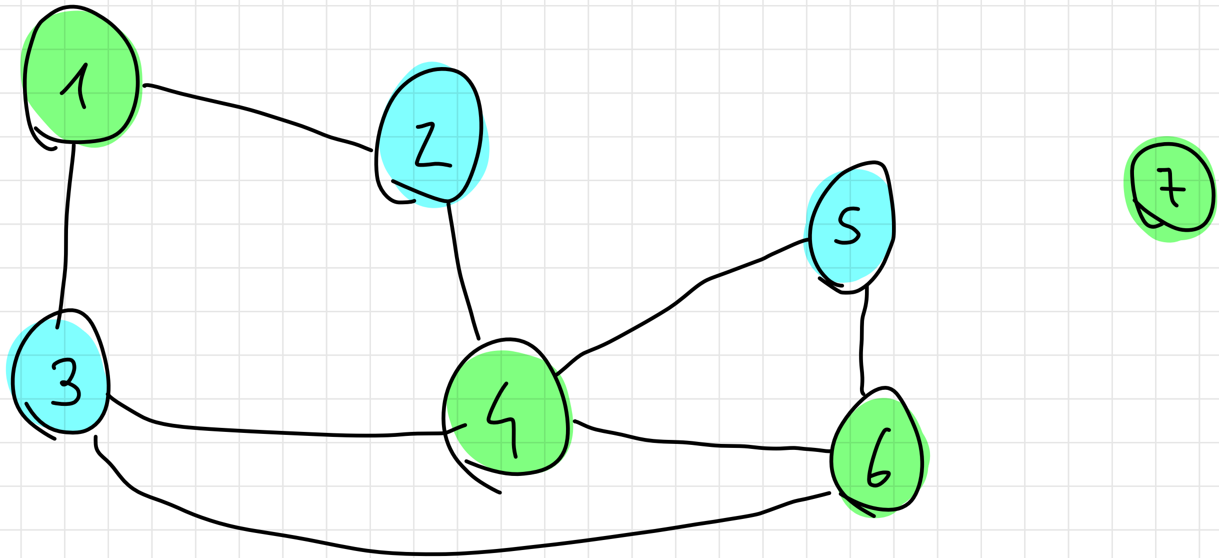
es.:

Ci sono n Pugili. C'è una rel. di rivalità
(assumiamo riflessiva)

Obiettivo: trovare una struttura dati per salvare questo insieme e partizionarlo in due team affinché non ci siano pugili rivali nello stesso team




→ grafo indiretto




Modifico BFS o DFS

es TdE:

$$L = \{ \underbrace{x . y . x^R} . \underbrace{x . y . x^R} \mid x \in \{a,b\}^+ , y \in \{a,b,c\}^* \}$$

Descrivere: MT k-nastri, RAM accettano L

Valutare compl. spaziali e temp. (RAM anche log.)

$x . y . x^R$: inizia e finisce con la stessa lettera $(a, b)$

$$\underbrace{aba}_{x} \underbrace{bcabccca}_{y \in \{a, b, c\}^*} \underbrace{aba}_{x^R}$$

MT k-nastri:

- Salvare la lungh. in unario in un nastro
- Controllare che $x \cdot y \cdot x^R$ abbia la stessa lettera $\in \{a,b\}$ a inizio e fine
- Controllare che la prima metà sia uguale alla seconda metà

$$T_{MT}(n) = \Theta(n)$$

$$S_{MT}(n) = \Theta(n)$$

$$a\underbrace{bcaccc}_{y}a \quad abcaccca$$

RAM :

a b c c a a b c c a

Salviamo a in mem.

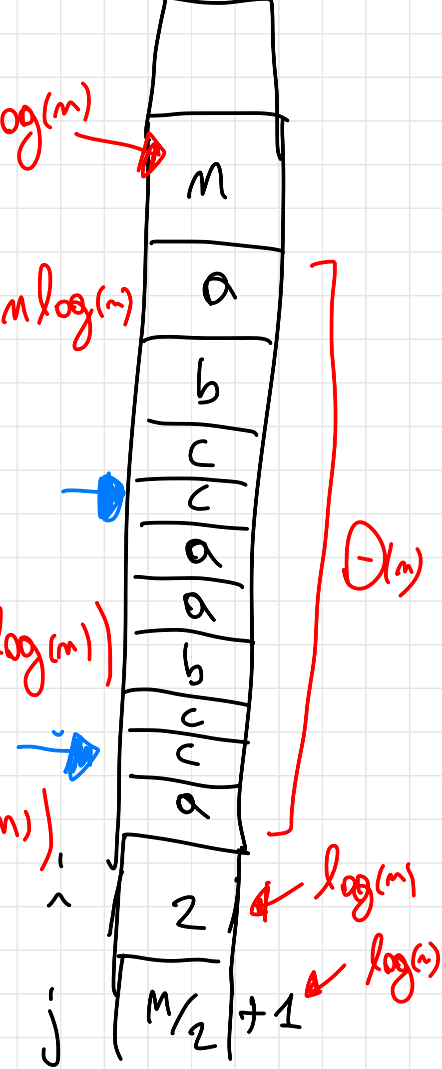
Salviamo la lung. in un contatore $\Theta(n)$

$\sum_{i=1}^{n} \log(i) \sim n \log(n)$

Uso due indici per confrontare le due semi-stringhe, controllando inizio e fine di una semi-stringa

$\Theta(n)$

$\Theta(n \log(n))$




$T^{COST}_{RAM}(n) = \Theta(n)$  $T^{log}_{RAM}(n) = \Theta(n \log(n))$

$S^{COST}_{RAM}(n) = \Theta(n)$  $S^{log}_{RAM}(n) = \Theta(n)$

es. TdE

$$g(y) = \begin{cases} y+1 & \text{se } f_y(y) = y \\ f_y(y) & \text{else} \end{cases}$$

$f_y \equiv$ f. calcolata da $MT_y$

g calcolabile?

||

esiste un algoritmo che la calcola?

g totale?

Provo a calcolare $g(y)$:

Calcolo $f_y(y)$:

se termina e $f_y(y) = y$:
return $y+1$

se termina e $f_y(y) \neq y$:
return $f_y(y)$

se non termina?
$\perp$

**Esercizio 3 (punti 6/15)**

1. Si consideri la funzione $g(y) = \text{if } f_y(y) = y \text{ then } y+1 \text{ else } f_y(y)$. Si dica se g è totale e se è calcolabile, motivando adeguatamente la risposta.
2. Sia A un insieme semidecidibile e B un insieme decidibile. E' facile verificare che l'insieme $C = A - B$ è un insieme semidecidibile: si fornisca all'uopo una funzione calcolabile (eventualmente anche parziale) che abbia come immagine C.

**Esercizio 3**

1) g è parziale e calcolabile: basta simulare la MT y con y in ingresso. Se essa termina, si controlla il risultato t e, se esso è y, si somma 1; altrimenti va bene t, anche se non dovesse essere definito.

2) $g_C(x) = \text{if } c_B(g_A(x)) = 0 \text{ then } g_A(x) \text{ else } \perp$

Se A semi-dec., allora $\mathbb{1}'_A(x) = \begin{cases} 1 & x \in A \\ \perp & x \notin A \end{cases}$ calcolabile

Se B dec., allora $\mathbb{1}_B(x) \left(= c_B(x)\right) = \begin{cases} 1 & x \in B \\ 0 & x \notin B \end{cases}$ calcolabile

Allora $g_A(x) = \begin{cases} x & \text{se } \mathbb{1}'_A(x) = 1 \\ \perp & \text{altrimenti} \end{cases}$ $\Rightarrow$ banalmente calcolab. usando $\mathbb{1}'_A$.

Quindi, posso calcolare $g_C(x)$ come mostrato in soluzione a partire da $g_A(x)$ e $c_B(x)$. $g_C(x)$ ha come immagine l'insieme $C$.