

Esercizio

La *teoria dei 6 gradi di separazione* fu formulata per la prima volta nel 1929 dallo scrittore ungherese Frigyes Karinthy nel racconto omonimo pubblicato nel volume *Catene*.

- 1 Trovare l'enunciato di tale teoria.
- 2 Definire un problema computazionale su grafo tale che la sua risoluzione possa servire a confermare o confutare la teoria, usando Facebook come insieme di dati di input.
- 3 Progettare e analizzare un algoritmo efficiente per il problema definito nel punto precedente.

① Ogni persona può essere collegata a una qualsiasi altra persona attraverso una catena di amicizie con ≤ 5 intermediari.

② $G = (V, E)$ graf d Facebook

$V = \{\text{profil.}\}$

$E = \{\text{amicizie}\}$

Assumiamo G connesso (se non lo è, lo è verificabile in tempo $O(n+m)$ chiaramente la teoria è confutata)

Se G è connesso la teoria sarà vera se
 $\forall v, u \in V$ la lunghezza del cammino
minimo tra v e u è ≤ 6 , e quindi se

$\forall v \in V$ la BFS (G, v) usa i livelli L_i
non vuoti solo per $i \leq 6$

- ③ Modifichiamo BFS in modo che restituisca
l'indice dell'ultimo livello L_i non vuoto
(le modifiche non ne alterano le complessità)

Algorithm NextSeparator (G)

input $G = (V, E)$ graph of Facebook (17. connes)

output $\max_{u, v \in V} \text{distance}(u, v)$

→ lunghezza del cammino
minimo

$\text{max_sep} \leftarrow 0;$

for all $v \in V$ do {

for all $u \in V$ do $u.ID \leftarrow 0;$

for all $e \in E$ do $e.\text{label} \leftarrow \text{null};$

$\text{max_sep} \leftarrow \max \{ \text{max_sep}, \text{BFS}(G, v) \}$

} ↘ modify

if ($\text{max_sep} \leq 6$) then return "TEORIA CONFIRMATA"

else return "TEORIA CONFUTATA"

Completezza :

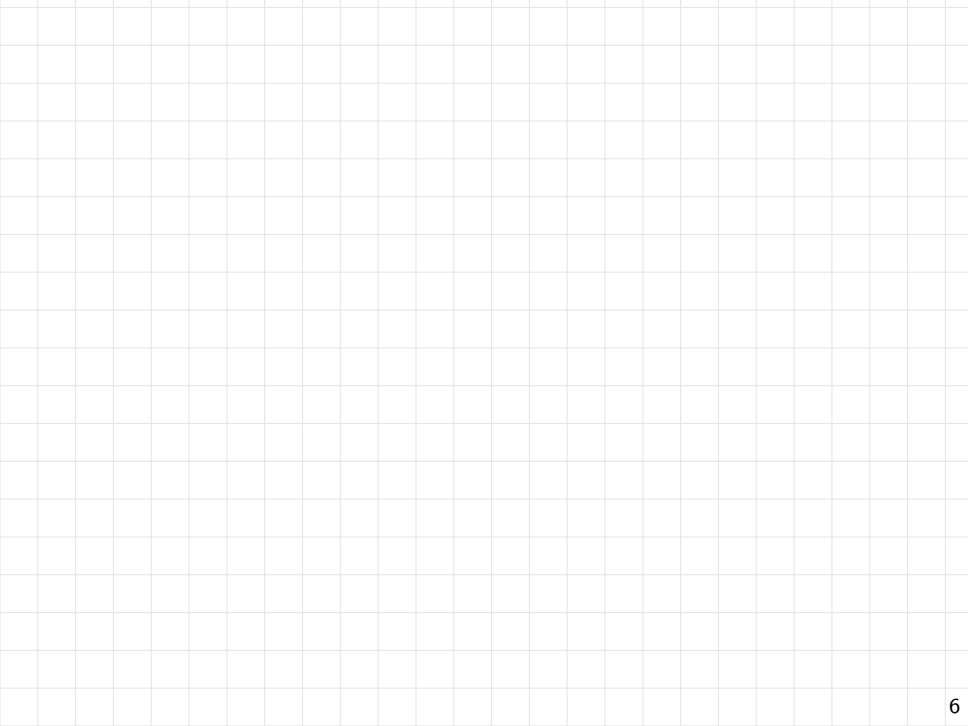
* Ogni iterazione del primal esterno costa

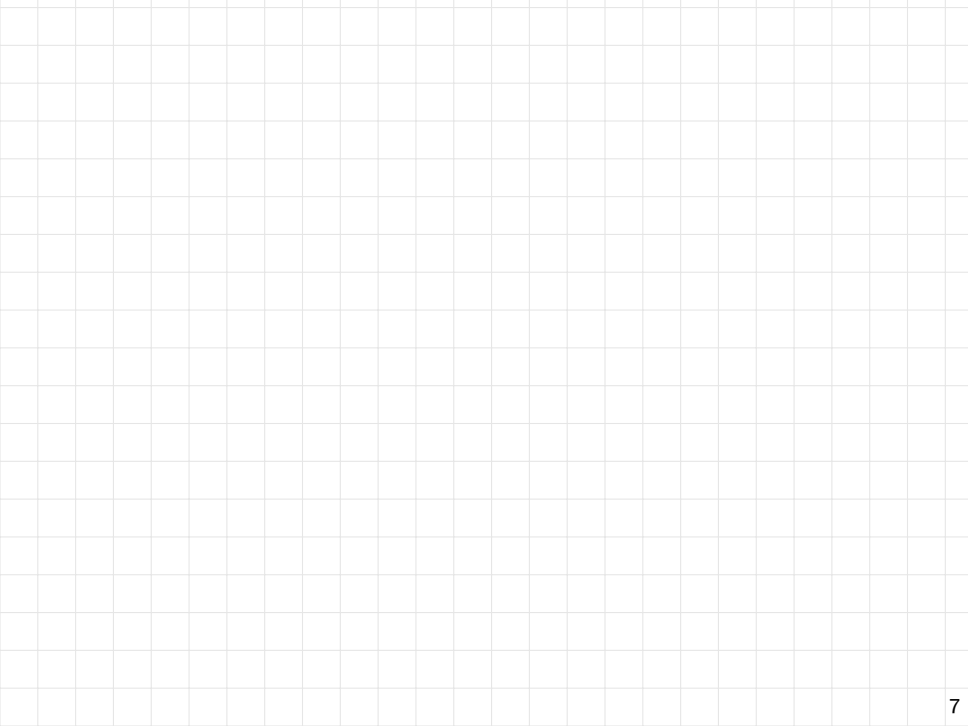
$\Theta(n+m) = \Theta(m)$ dato che assumiamo

G connected

* n iterazioni del primal esterno

$\Rightarrow$ Totale : $\Theta(n \cdot m)$





Esercizio C-14.49 [GTG14]

Un grafo $G = (V, E)$ si dice *bipartito* se l'insieme di vertici V può essere partizionato in due sottoinsiemi X e Y tali che ogni arco di E incide su un vertice di X e uno di Y . Progettare e analizzare un algoritmo efficiente che determini se un grafo non diretto G è bipartito.

Concludiamo il caso in cui G è connesso (il caso più generale è lasciato come ESERCIZIO)

IDEA: Eseguiamo $\text{BFS}(G, s)$ da un vertice $s \in V$ arbitrario. Si ha che V viene partizionato in

LIVELLI : $L_0 = \{s\}, L_1, L_2, \dots, L_i \dots$

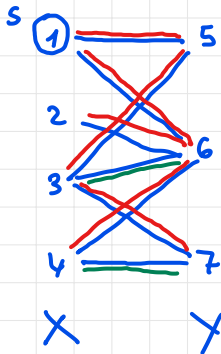
Tali che $\forall i > 0, L_i$ contiene tutti e solo i figli

di L_{i-1} nell'albero de. DISCOVERY EDGE

Quindi, se G è BIPARTITO (con $V = X \cup Y$)
deve valere PERFORA che i livelli d'indice
par. contengono tutti e soli i vertici di uno de.
2 insiem., X o Y , e i livelli d'indice dispar.
quell. dell'altro insieme.

$\Rightarrow G$ sarà BIPARTITO se ogni CROSS EDGE (u, v)
ha u e v in livelli adiacenti e non nello stesso
livello

ESEMPIO



$$L_0 = 1$$

$$L_1 = 5 \ 6$$

$$L_2 = 3 \ 2 \ 4$$

$$L_3 = 7$$

In rosso i DISCOVERY EDGE

In verde i CROSS EDGE

Modifica a $BFS(G, s)$: $\forall$ ogni vertice $v \in L_i$ c'è
un camp $v.level$ che viene impostato a i

Algorithm is Bipartite (G)

input $G = (V, E)$

output TRUE/FALSE if G is / is not bipartite

for all $v \in V$ do $\{v.id \leftarrow 0; v.level \leftarrow null\}$

for all $e \in E$ do $e.label \leftarrow null$

for $s \in V$ arbitrary

BFS(G, s)

for all $e = (u, v) \in E$ do

if $(u.level = v.level)$ then return FALSE

```
}  
return TRUE
```

Complessità : $\Theta(n+m) = \Theta(m)$ dato
che G è connesso

Esercizio: generalizzare l'algoritmo al
caso di G non connesso.

Esercizio

Sia $G = (V, E)$ un grafo non diretto con $k > 1$ componenti connesse. Progettare un algoritmo basato sulla DFS che aggiunga $k - 1$ archi a G per renderlo connesso, e analizzarne la complessità. Si assuma di poter aggiungere a E un arco $(u, v) \notin E$ in tempo costante invocando il metodo $G.\text{addArc}(u, v)$.

