17. Algoritmi 3 – Ricerca in ampiezza

Corso di Informatica

Corso di Laurea in Matematica (D.M. 270/04) - A.A. 2020/2021
Angelo Cardellicchio
angelo.cardellicchio@uniba.it
11/11/2020

Outline

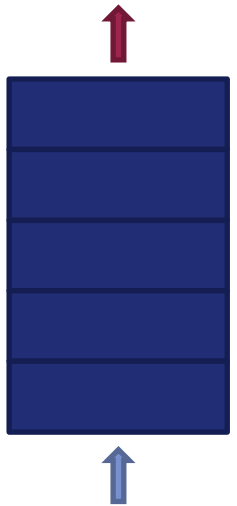
- Ricerca in ampiezza
  - Alberi
  - Grafi
  - Analisi computazionale

Ricerca in ampiezza

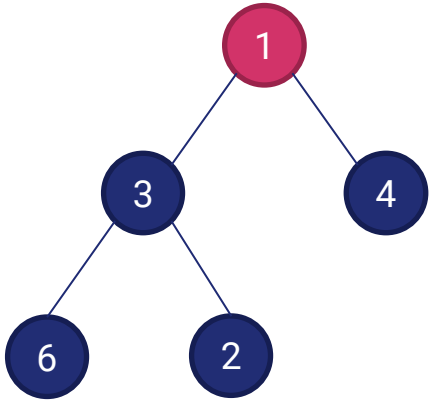
- Nota anche come **Breadth – First Search** (**BFS**)
- Lavora su componenti connesse, indipendentemente dal fatto che siano grafi od alberi
  - *Opera attraversando tutti i nodi presenti alla stessa distanza da un nodo sorgente, e quindi proseguendo al livello successivo*
- Può essere implementata mediante una coda
- Applicazioni:
  - *Trovare persone a k collegamenti di distanza*
  - *Individuare nodi adiacenti di una rete*
  - *Trovare il cammino minimo tra due nodi*

Ricerca in ampiezza – Alberi

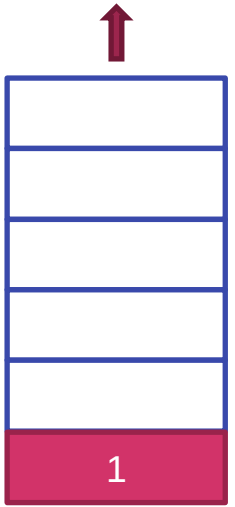
```
STEP 1: create(queue); current_node = root;
STEP 2: while (current_node is not null)
STEP 3:     get(current_node_children);
STEP 4:     enqueue(queue, current_node_children);
STEP 5:     current_node = dequeue(queue)
```



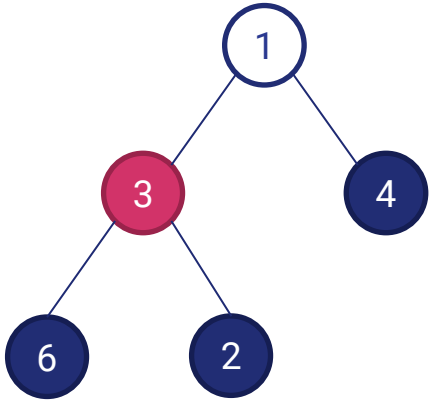
Ricerca in ampiezza – Alberi



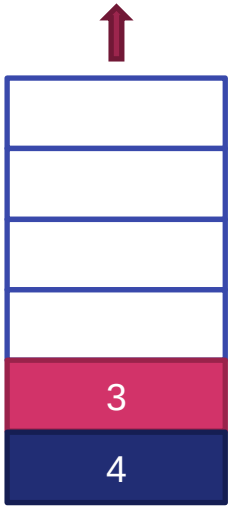

```
current_node = {1}
children = {3, 4}
```



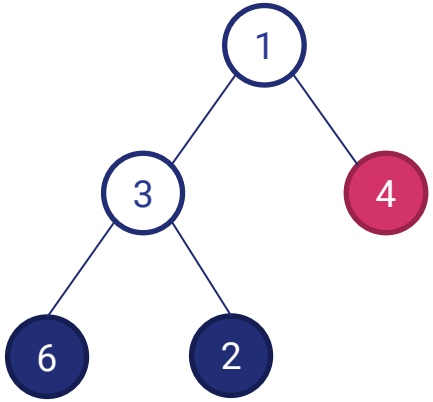
Ricerca in ampiezza – Alberi



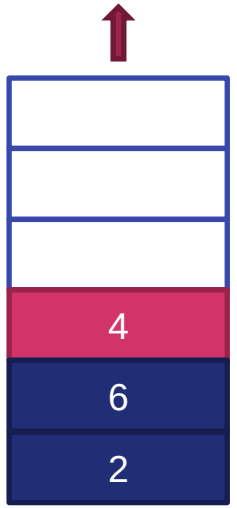

```
current_node = {3}
children = {6, 2}
dequeued = {1}
```



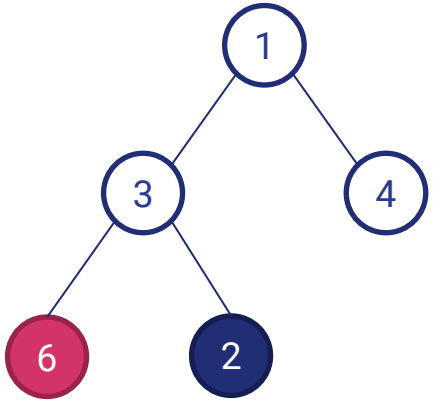
Ricerca in ampiezza – Alberi



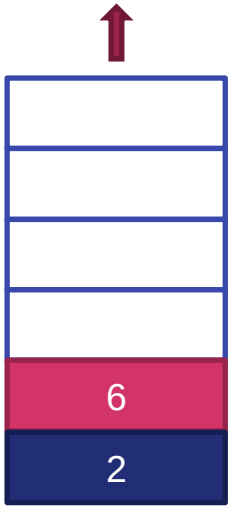

```
current_node = {4}
children = {}
dequeued = {1, 3}
```



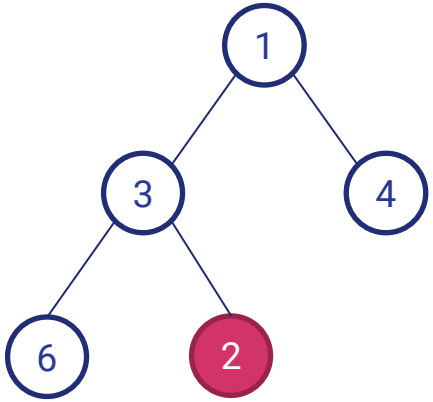
Ricerca in ampiezza – Alberi



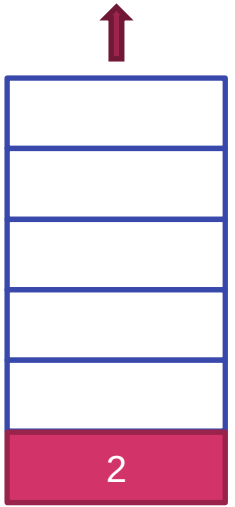

```
current_node = {6}
children = {}
dequeued = {1, 3, 4}
```



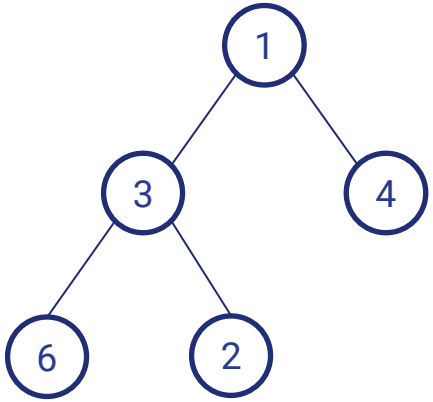
Ricerca in ampiezza – Alberi



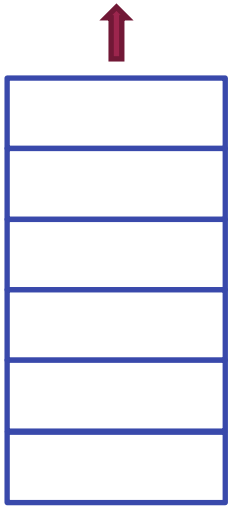

```
current_node = {2}
children = {}
dequeued = {1, 3, 4, 6}
```



Ricerca in ampiezza – Alberi

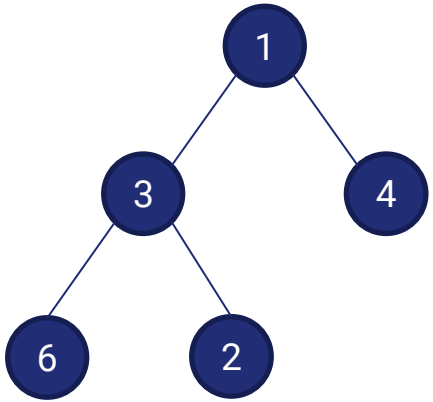


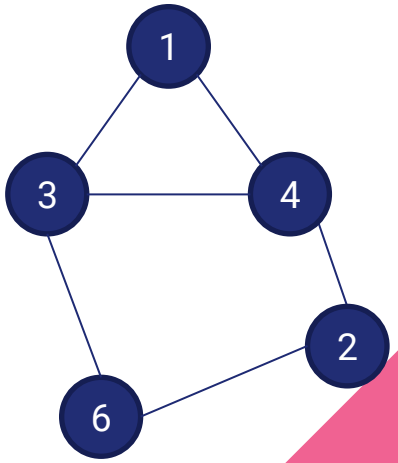

```
current_node = {}
children = {}
dequeued = {1, 3, 4, 6, 2}
```



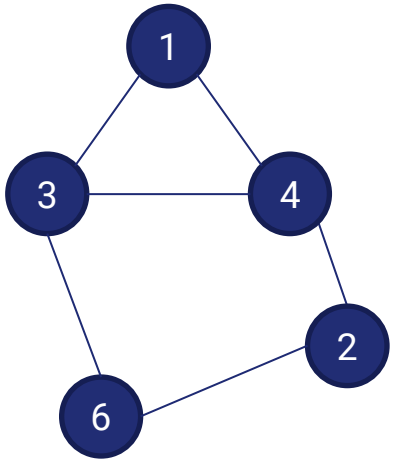
Ricerca in ampiezza – Alberi vs. Grafi

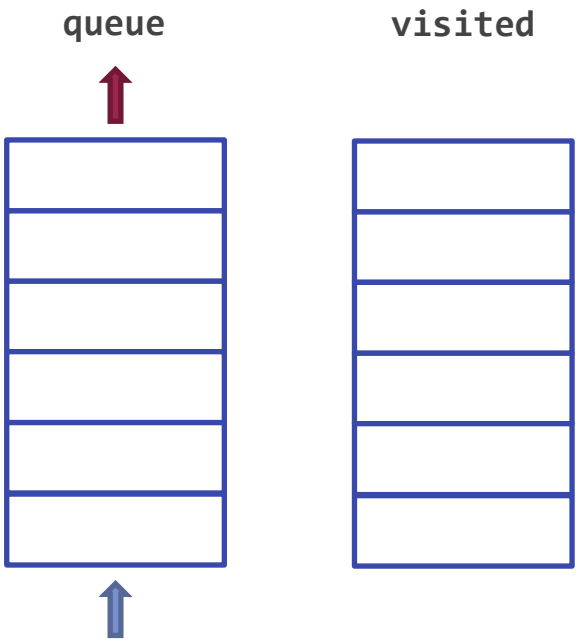
- La differenza principale tra alberi e grafi è che i primi non hanno cicli, i secondi invece sì
  - *Occorre quindi tener conto dei **nodi già visitati***
  - *Per farlo, usiamo un apposito array*





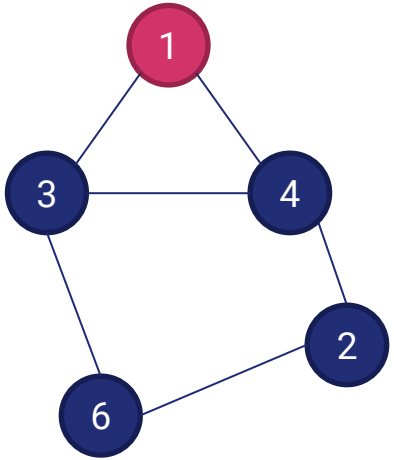
Ricerca in ampiezza – Grafi

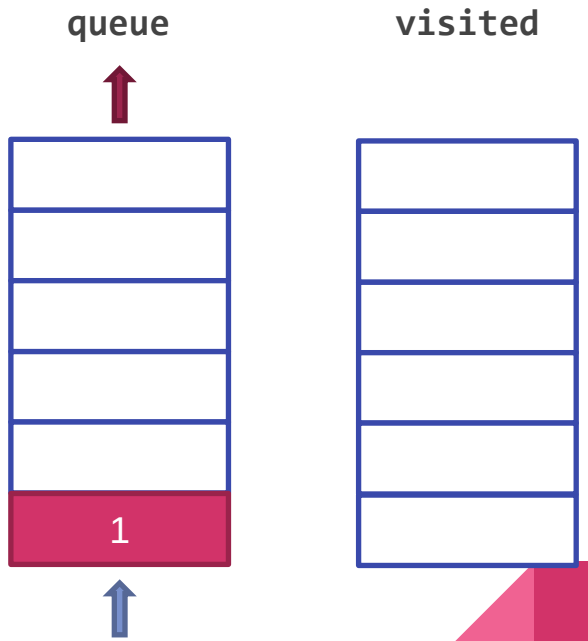




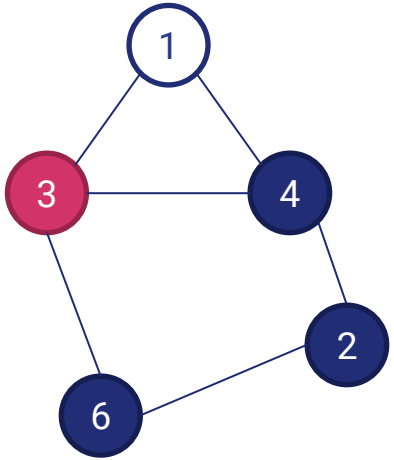

```
if (node is not in visited)
    visit(node)
```

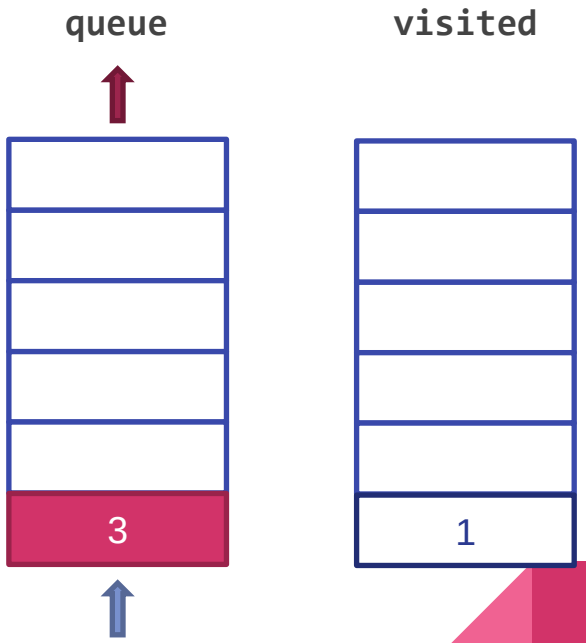
Ricerca in ampiezza – Grafi



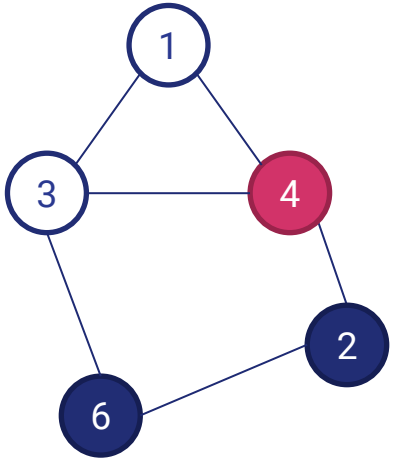


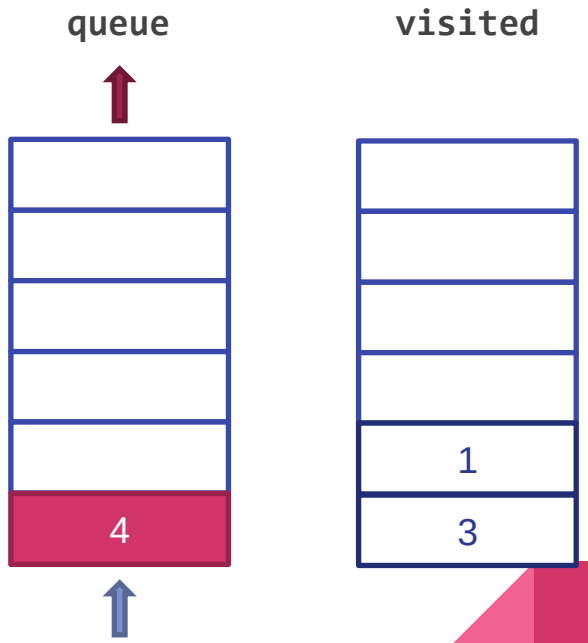
Ricerca in ampiezza – Grafi



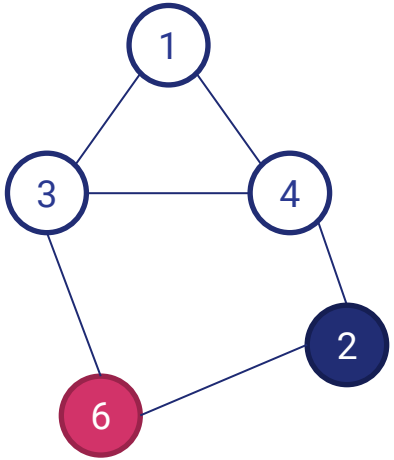


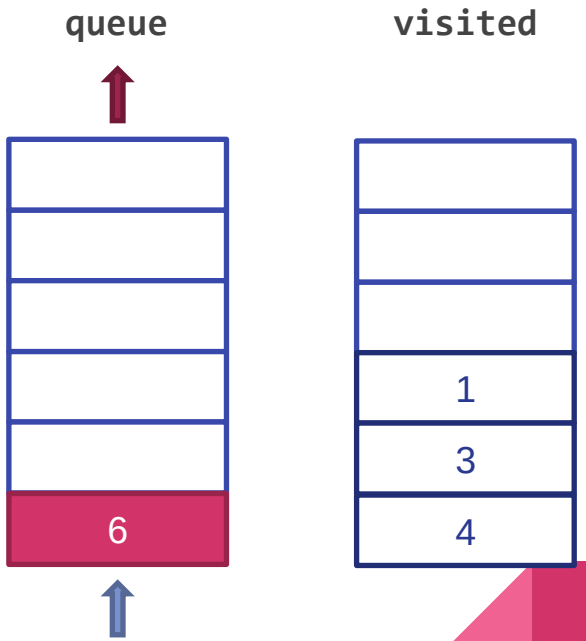
Ricerca in ampiezza – Grafi



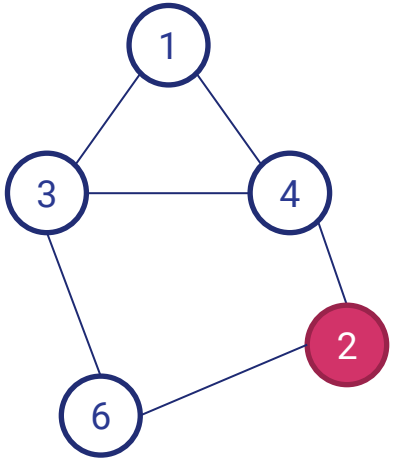


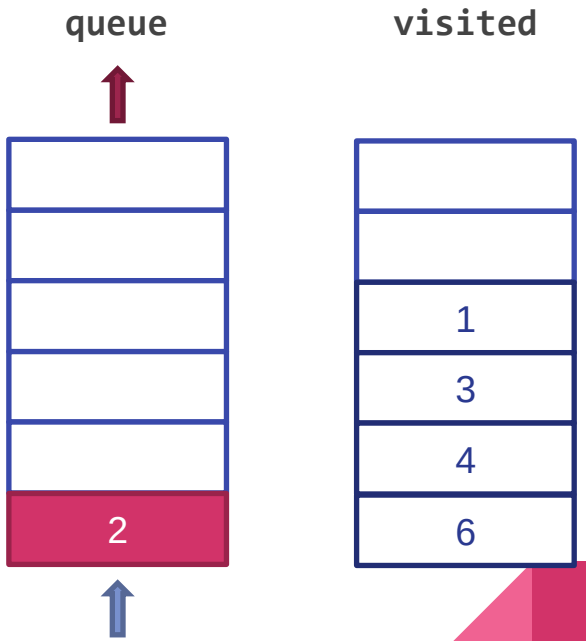
Ricerca in ampiezza – Grafi



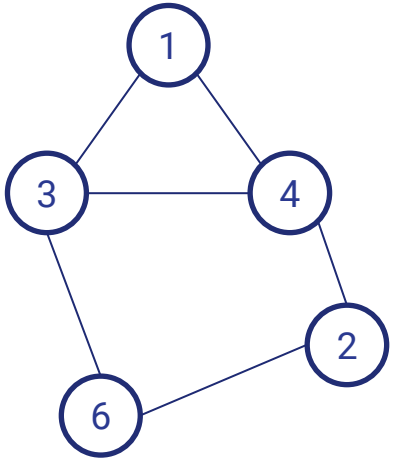


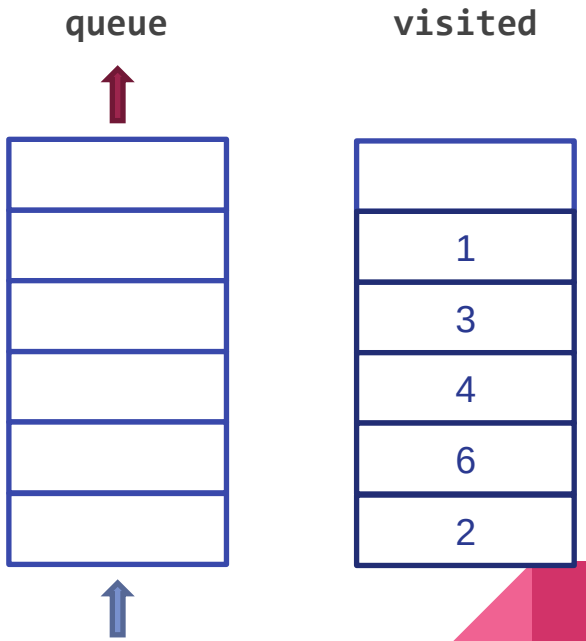
Ricerca in ampiezza – Grafi





Ricerca in ampiezza – Grafi





Ricerca in ampiezza – Analisi computazionale

- Supponiamo di avere n elementi nel nostro albero/grafo
- La ricerca in ampiezza dovrà esaminare al più n elementi (nel caso sia implementata come una coda)
- Considerando costante il costo computazionale legato all'eventuale estrazione dei nodi figli (c_F) ed alle operazioni legate alla coda (c_Q), potremo dire che il costo complessivo è di $(c_F + c_Q) \cdot n$
- La complessità computazionale sarà quindi in $O(n)$

Domande?

42