



Università di Roma



# AVL Tree

*Romolo Marotta*

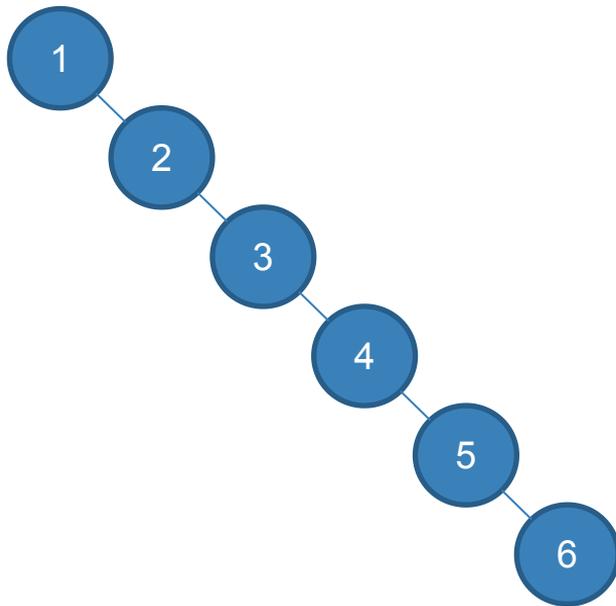
*marotta@diag.uniroma1.it*

*Alessandro Pellegrini*

*pellegrini@diag.uniroma1.it*

# Alberi binari di ricerca degeneri

- Le operazioni sugli alberi binari di ricerca hanno costo  $O(h)$
- $h$  dovrebbe essere tipicamente pari a  $\log n$ , ma una sequenza sfavorevole di inserimenti può generare una situazione degenera in cui  $h = n$

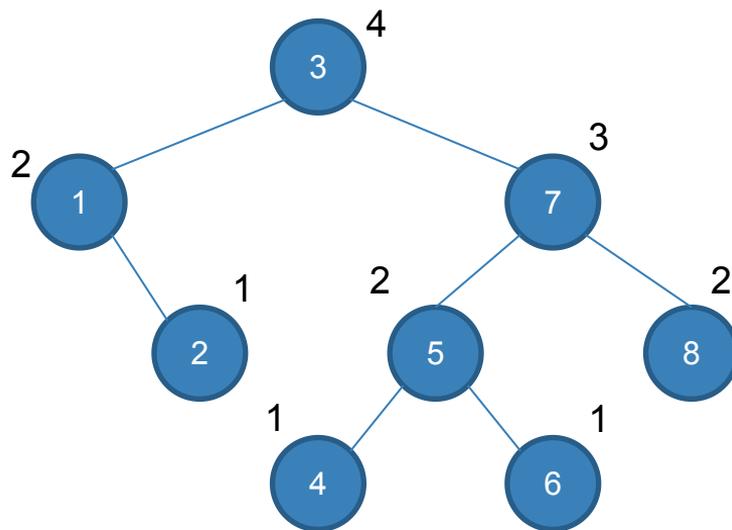


# Alberi binari di ricerca auto-bilanciati

- La probabilità che, dati  $n = 2^k$  nodi, un albero binario di ricerca sia completo e pieno è molto piccola
  - ▶ Sono poche le permutazioni degli inserimenti che generano un albero completamente bilanciato
- Soluzione: una struttura dati adattativa che “reagisce” allo sbilanciamento riorganizzando la sua struttura interna
  - ▶ Gli alberi auto-bilanciati mescolano i nodi precedentemente inseriti, senza violare la proprietà dell'albero binario di ricerca

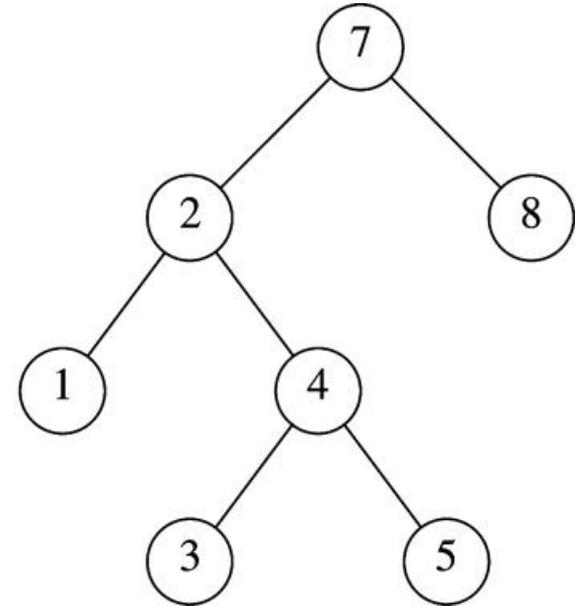
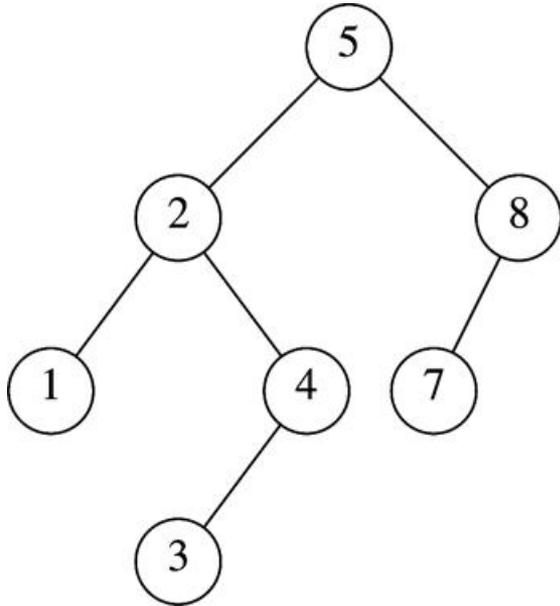
# AVL Tree

- Gli alberi AVL (Adelson-Velskii and Landis) sono degli alberi auto-bilanciati in cui, per ogni nodo interno  $v$ , la differenza di altezza tra i figli destro e sinistro di  $v$  può essere al più 1



# AVL Tree

- Quale di questi è un albero AVL?

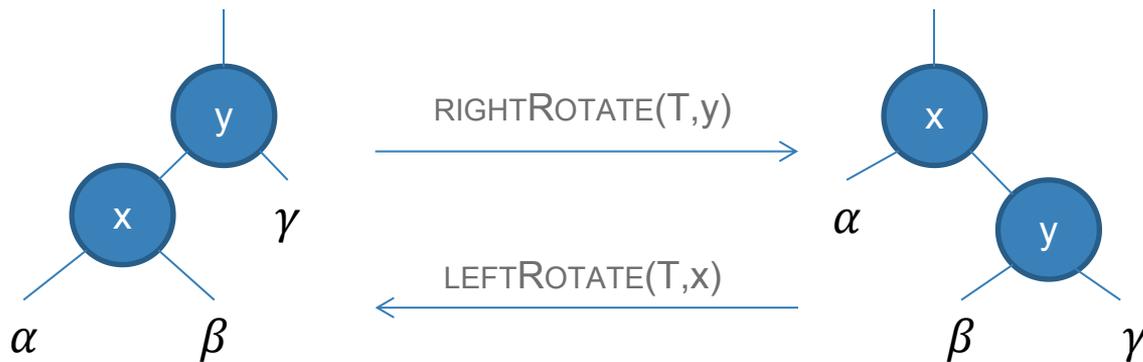


# Inserimenti ed eliminazioni

- L'inserimento e l'eliminazione avvengono secondo gli algoritmi `TREEINSERT()` e `TREEDELETE()` propri degli alberi binari di ricerca
- Come nel caso degli alberi RB, questo inserimento può portare ad una violazione della *condizione di bilanciamento*
  - ▶ Negli alberi RB: il numero di nodi neri in ogni percorso radice/foglia è lo stesso
  - ▶ Negli alberi AVL: la differenza di altezza dei due figli è al più 1
- Azione correttiva: *ribilanciamento*, per ripristinare la condizione

# Operazione fondamentale: la rotazione

- Le operazioni di “rotazione” sono operazioni locali che conservano la proprietà fondamentale degli alberi binari di ricerca
- Le rotazioni coinvolgono una coppia di nodi e tutti i loro sottoalberi destro e sinistro

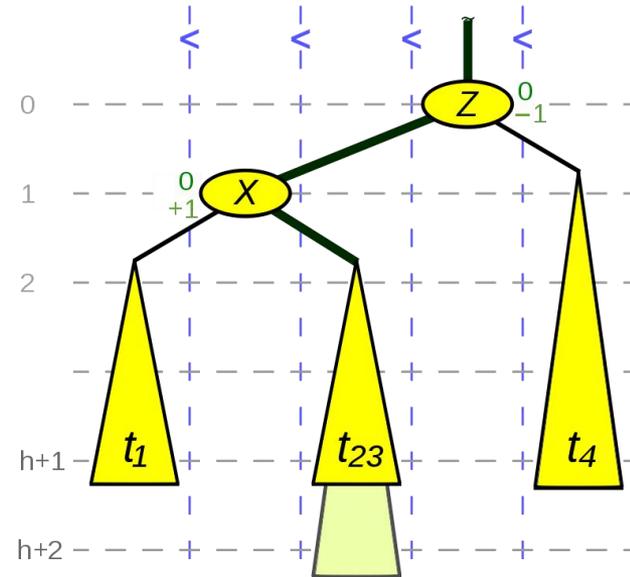
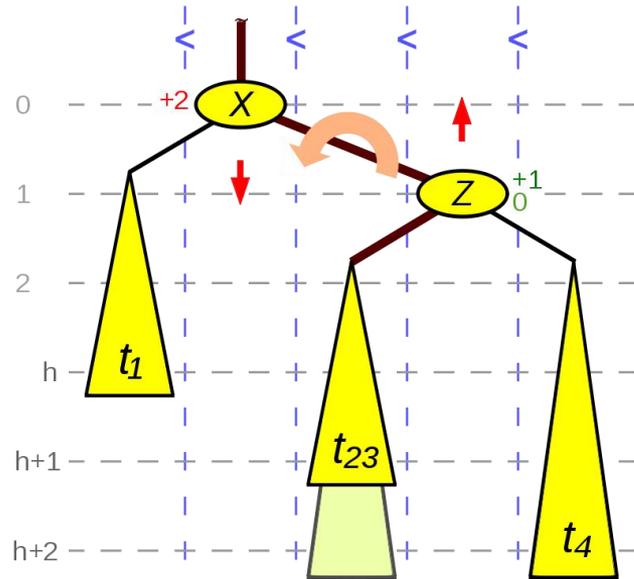


# Quale parte dell'albero ribilanciare?

- Nel caso di violazione dopo un inserimento:
  - ▶ Solo i nodi nel percorso foglia-radice coinvolto nell'inserimento possono osservare una violazione della condizione di bilanciamento
  - ▶ Si ribilancia il nodo *più profondo* in cui si osserva uno sbilanciamento (l'intero albero viene così ribilanciato)
- Quattro casi di sbilanciamento da gestire (nel nodo più profondo k):
  1. Inserimento nel sottoalbero sx del figlio sx di k
  2. Inserimento nel sottoalbero dx del figlio sx di k
  3. Inserimento nel sottoalbero sx del figlio dx di k
  4. Inserimento nel sottoalbero dx del figlio dx di k

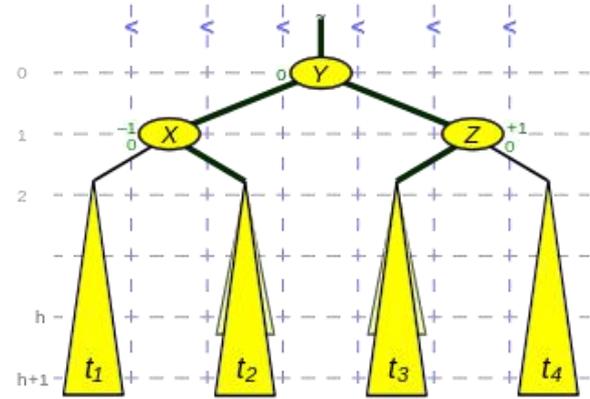
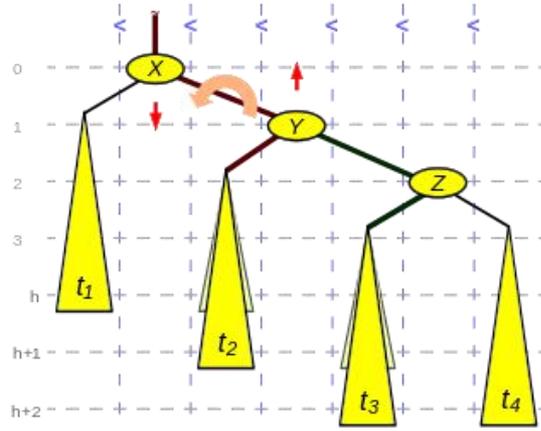
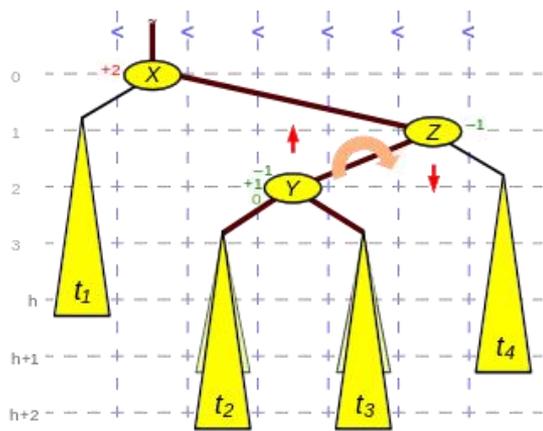
# Che rotazione effettuare?

- I casi 1 e 4 sono equivalenti (simmetrici)
- I casi 2 e 3 sono equivalenti (simmetrici)
- Caso 1: **rotazione singola**



# Che rotazione effettuare?

- I casi 1 e 4 sono equivalenti (simmetrici)
- I casi 2 e 3 sono equivalenti (simmetrici)
- Caso 2: **rotazione doppia** (la prima rotazione riduce i casi 2/3 ai casi 1/4)



# Un esempio in pratica

