Algoritmi e Strutture Dati

Esercitazione 5

Domenico Fabio Savo

Esercitazione: alberi AVL

Esercizio:

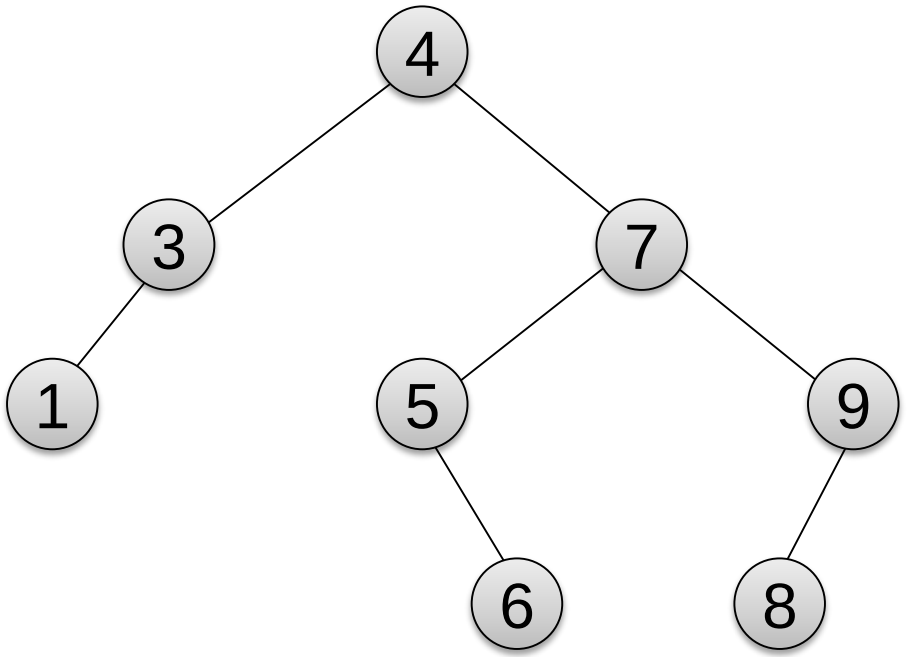
1. Costruire un albero AVL contenente le seguenti chiavi:

4 3 1 7 6 5 9 8

Esercitazione: alberi AVL

Soluzione:

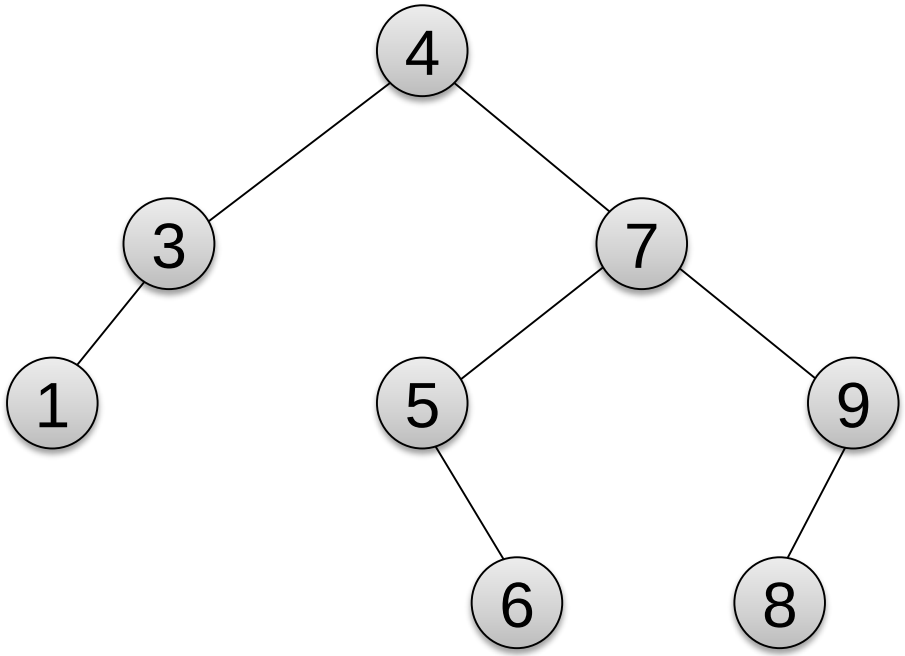
Una possibile albero AVL è il seguente



Esercitazione: alberi AVL

Esercizio:

2. Descrivere le operazioni (rotazioni effettuate, fattori di bilanciamento e nodi critici di ogni passo) da effettuare se rimuoviamo la radice.

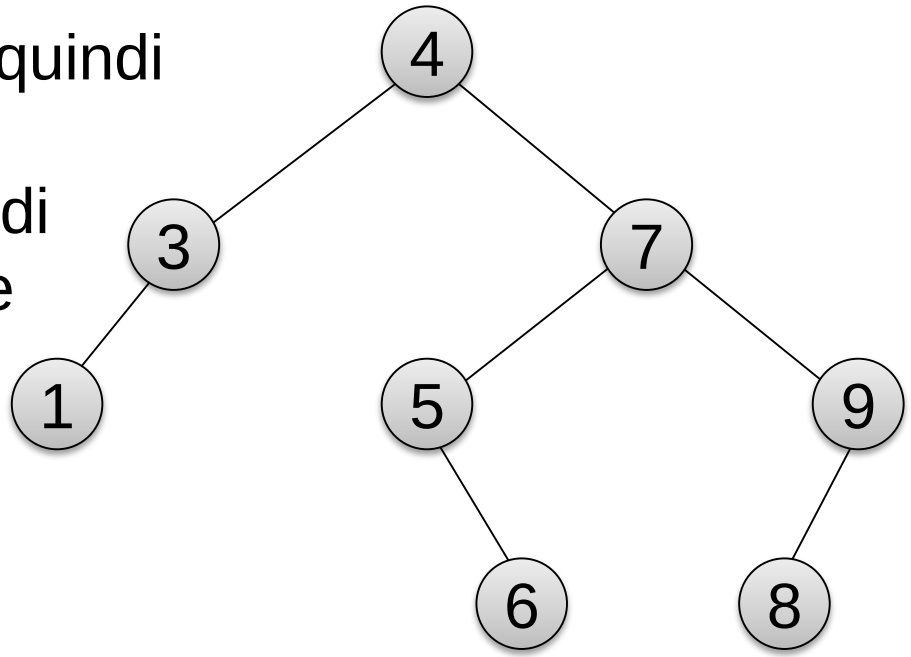


Esercitazione: alberi AVL

Soluzione:

2 – La cancellazione avviene come per i BST

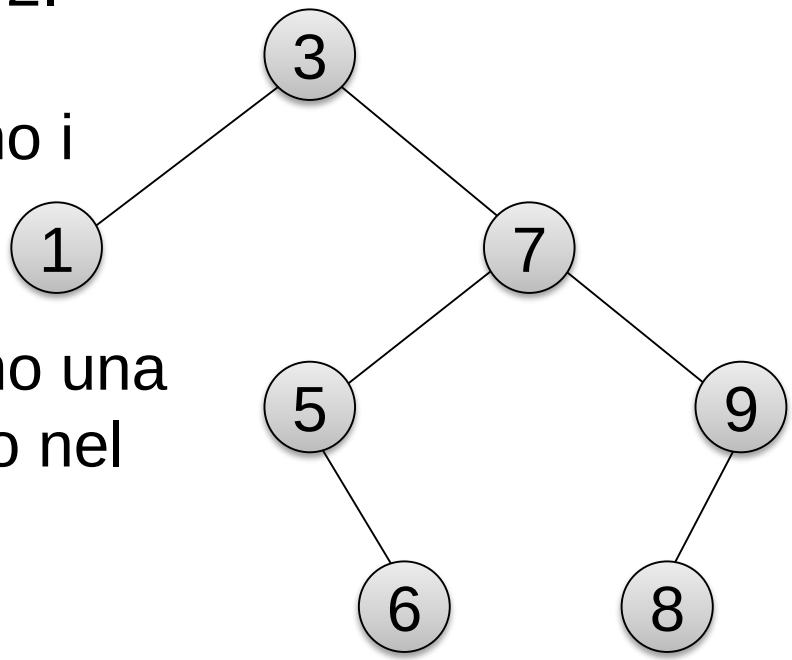
- La radice ha due figli, siamo quindi nel CASO 3;
- Il predecessore di 4 è 3, quindi il nodo 3 sarà la nuova radice



Esercitazione: alberi AVL

Soluzione:

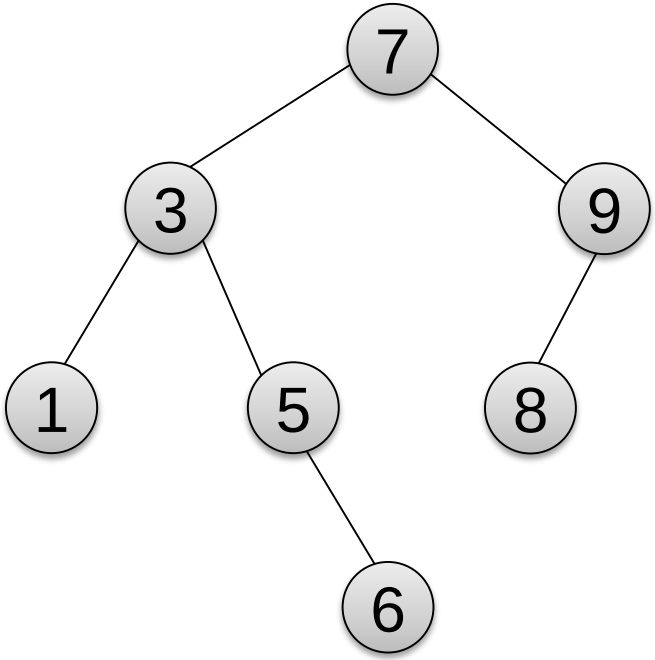
- L'albero ottenuto è sbilanciato: il nodo 3 (nodo critico) ha un fattore di bilanciamento pari a 2.
- A generare lo sbilanciamento sono i sotto alberi del nodo 7.
- Per ribilanciare l'albero effettuiamo una rotazione verso sinistra con perno nel Nodo 3.



Esercitazione: alberi AVL

Soluzione:

- Dopo la rotazione l'albero è nuovamente bilanciato



Esercizio – stampaDFS-Sim

Esercizio:

Scrivere lo pseudocodice della funzione

```
stampaDFS-Sim(nodo r)
```

che stampa i nodi dell'albero radicato in **r** nell'ordine con cui questi sono visitati da una visita simmetrica dell'albero.

Esercizio – stampaDFS-Sim

Esercizio:

Scrivere lo pseudocodice della funzione

`stampaDFS-Sim(nodo r)`

che stampa i nodi dell'albero radicato in **r** nell'ordine con cui questi sono visitati da una visita simmetrica dell'albero.

Soluzione:

```
     algoritmo stampaDFS-Sim(nodo r)
1.       if (r = null) then return
2.       stampaDFS-Sim(figlio sinistro di r)
3.       stampa(r);
4.       stampaDFS-Sim(figlio destro di r)
```

Esercizio - Stampa

Esercizio:

Sia **C** un coda contenente i seguenti interi:

C = { 1 6 7 2 3 8 } dove 8 è l'elemento di testa.

Fornire la sequenza di interi stampata dal seguente algoritmo

```
algoritmo stampa(coda C)
  sia A un albero AVL vuoto;
  while(not C.isEmpty())
          A.insert(C.dequeue());
  stampaDFS-Sim(A.getRoot());
  //getRoot() restituisce la radice dell'albero A
```

Esercizio - Stampa

Soluzione:

- Il ciclo **while** termina quando l'ultimo numero contenuto nella coda **C** viene inserito nell'albero **A**.

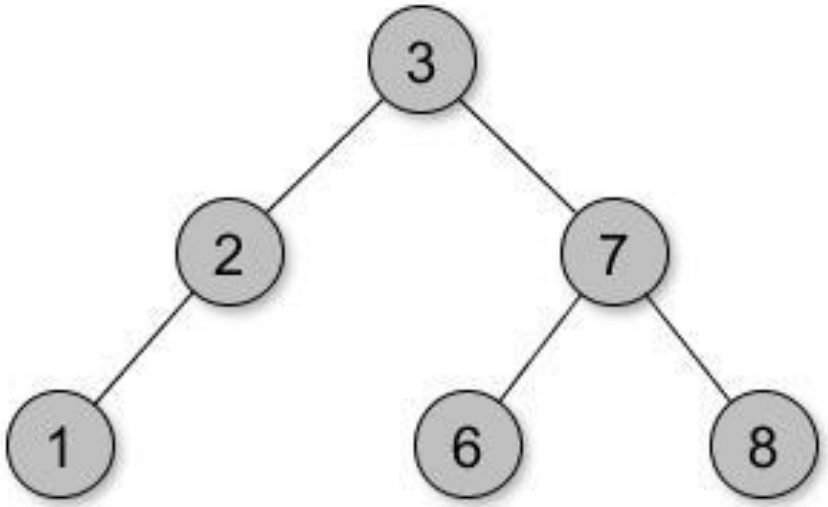
- Dato che **C** è una coda, allora gli elementi verranno inseriti in **A** a cominciare dall'elemento di testa (l'8) per poi proseguire verso l'elemento di coda (l' 1).

Esercizio - Stampa

Soluzione:

Al termine del ciclo **while** avremo il seguente albero AVL:

A =

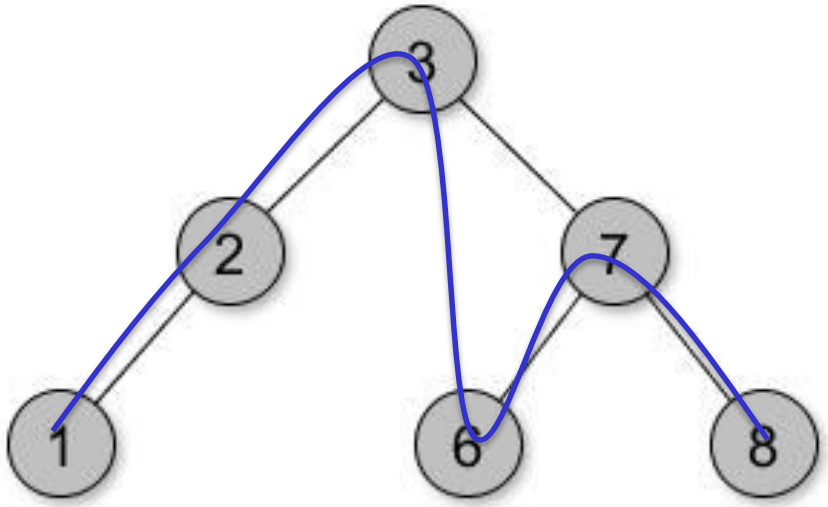


Esercizio - Stampa

Soluzione:

La funzione `stampaDFS-Sim(3)` effettuerà una visita simmetrica dell'albero e ad ogni visita stamperà la chiave del nodo visitato.
I nodi saranno stampati nel seguente ordine

1 – 2 – 3 – 6 – 7 – 8



Ordinamento e BST

È possibile ordinare un qualunque insieme di elementi X inserendo gli elementi in un **albero binario di ricerca** e poi eseguendo una visita simmetrica.

Domanda

Domanda:

```
algoritmo stampa(coda C)
  sia A un albero AVL vuoto;
  while(not C.isEmpty())
          A.insert(C.dequeue());
  stampaDFS-Sim(A.getRoot());
  //getRoot() restituisce la radice dell'albero A
```

- L'algoritmo `stampa(coda C)` può essere utilizzato per ordinare gli elementi contenuti in **C**. Quale è il tempo di esecuzione dell'algoritmo nel caso peggiore?
- E se invece di un AVL avessimo utilizzato un BST?