

ESAME DI ALGORITMI E STRUTTURE DATI
A.A. 2014/2015 – APPELLO STRAORDINARIO DEL 30/03/2015

TEMPO A DISPOSIZIONE: 120 MINUTI

ESERCIZIO 1:

- a) Si consideri una **CODA** rappresentata come una tripla **(L, h, t)** dove:
- **L** è una LISTA semplice in cui ogni record è una coppia (*elem, next*), dove *next* è il puntatore al successivo record della lista;
 - **h** è il puntatore al primo record della lista;
 - **t** è il puntatore all'ultimo record della lista.

Scrivere lo pseudocodice che implementa le seguenti quattro operazioni:

- isEmpty() → boolean
- enqueue(elem e)
- dequeue() → elem e
- first() → elem e

ESERCIZIO 2:

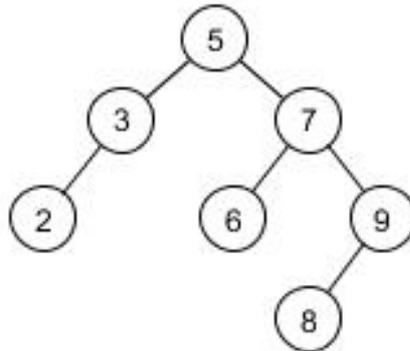
Mostrare il comportamento dell'algoritmo **RadixSort** per ordinare alfabeticamente le seguenti parole:

R A T A
B A R C A
A R T O
B A R
C A R T A

Notare che le parole hanno lunghezze diverse. Come risolviamo questo problema?

ESERCIZIO 3:

- a) Dare la definizione di "fattore di bilanciamento" di un nodo di un albero.
- b) Indicare le condizioni che un albero binario deve soddisfare per essere un albero AVL.



- c) Descrivere i passi da compiere se dall'albero AVL in figura rimuoviamo la radice indicando il tipo di rotazioni eventualmente effettuate ed il fattore di bilanciamento di eventuali nodi critici.

ESERCIZIO 4:

Fornire la definizione formale delle seguenti notazioni asintotiche:

$$f(n) = O(g(n))$$

$$f(n) = \Omega(g(n))$$

$$f(n) = \Theta(g(n))$$

ESERCIZIO 5:

- i) Si scriva lo pseudocodice dell'algoritmo `BubbleSort` in grado di ordinare gli elementi contenuti in un array **A** di **n** interi preso in ingresso.
- ii) Descrivere i passi eseguiti dall'algoritmo `BubbleSort` sul seguente array in input.

0	1	2	3	4	5	6	7
8	9	5	2	4	7	1	3

- iii) Indicare la complessità dell'algoritmo `BubbleSort` nel caso peggiore.