Corso di Programmazione e Strutture Dati
Docente di Laboratorio: Marco Romano
Email: marromano@unisa.it

ADT ALBERO BINARIO

ESERCIZI

1. Realizzare delle funzioni per determinare **l'altezza** e il **numero di nodi** di un albero binario

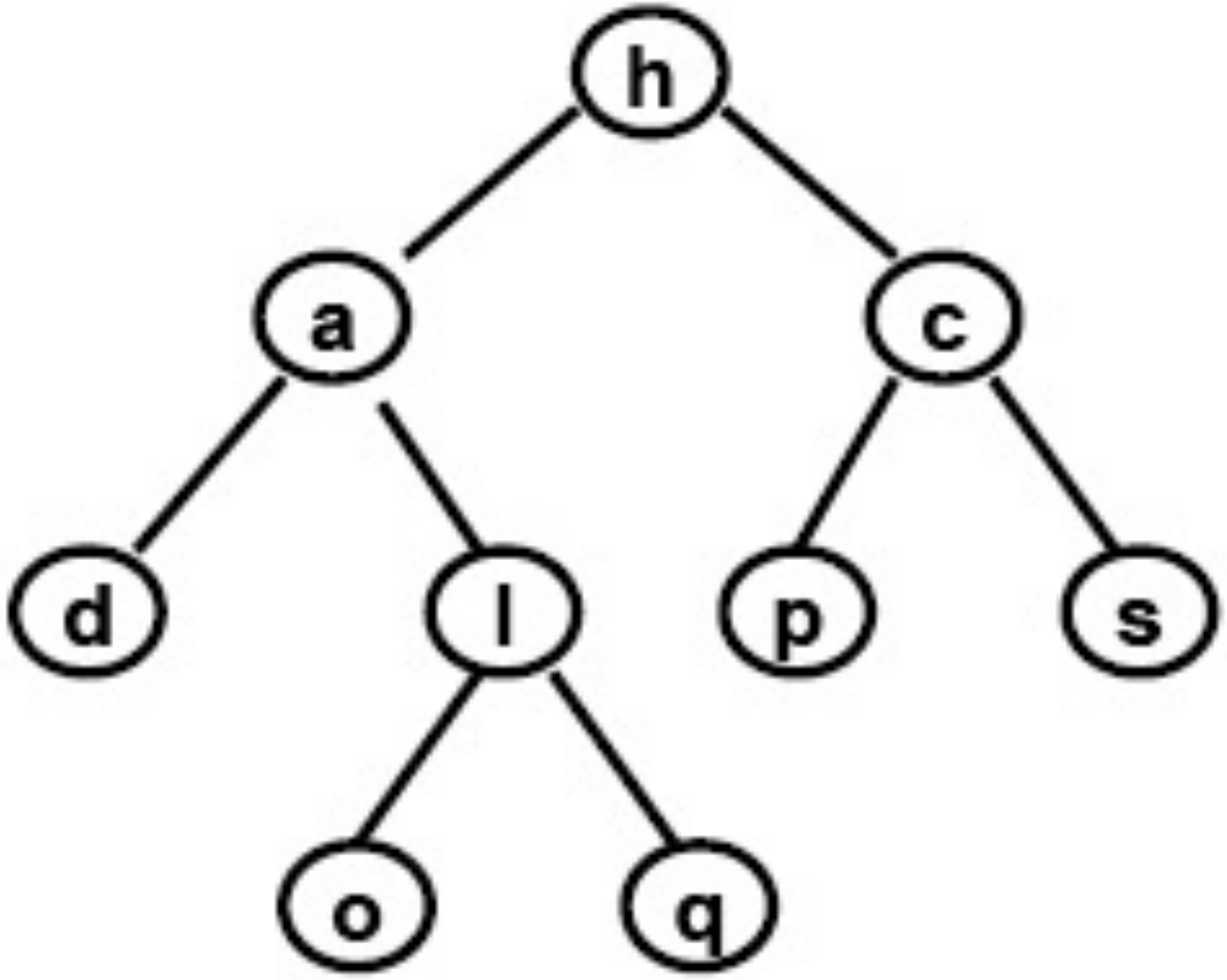
2. Realizzare una visita per livelli di un albero binario

3. Realizzare le tre visite dell'albero binario in maniera **iterativa,** con l'uso di uno **stack**

ESERCIZIO 1

Realizzare una o più funzioni per determinare **l'altezza di un albero** e una o più funzioni per determinare il **numero di nodi** di un albero binario.

L'altezza è il cammino più lungo partendo dalla radice



Numero nodi: 9
Altezza: 3

ESERCIZIO 1
PARTE 1

```
75  int max(int a, int b){
76      return a>b ? a : b;
77  }
78
79  int height(BTree bt){
80      if (isEmptyTree(bt))
81          return 0;
82      else if (!getLeft(bt) && !getRight(bt))
83          return 0;
84      else
85          return 1+max(height(bt->left),height(bt->right));
86  }
```

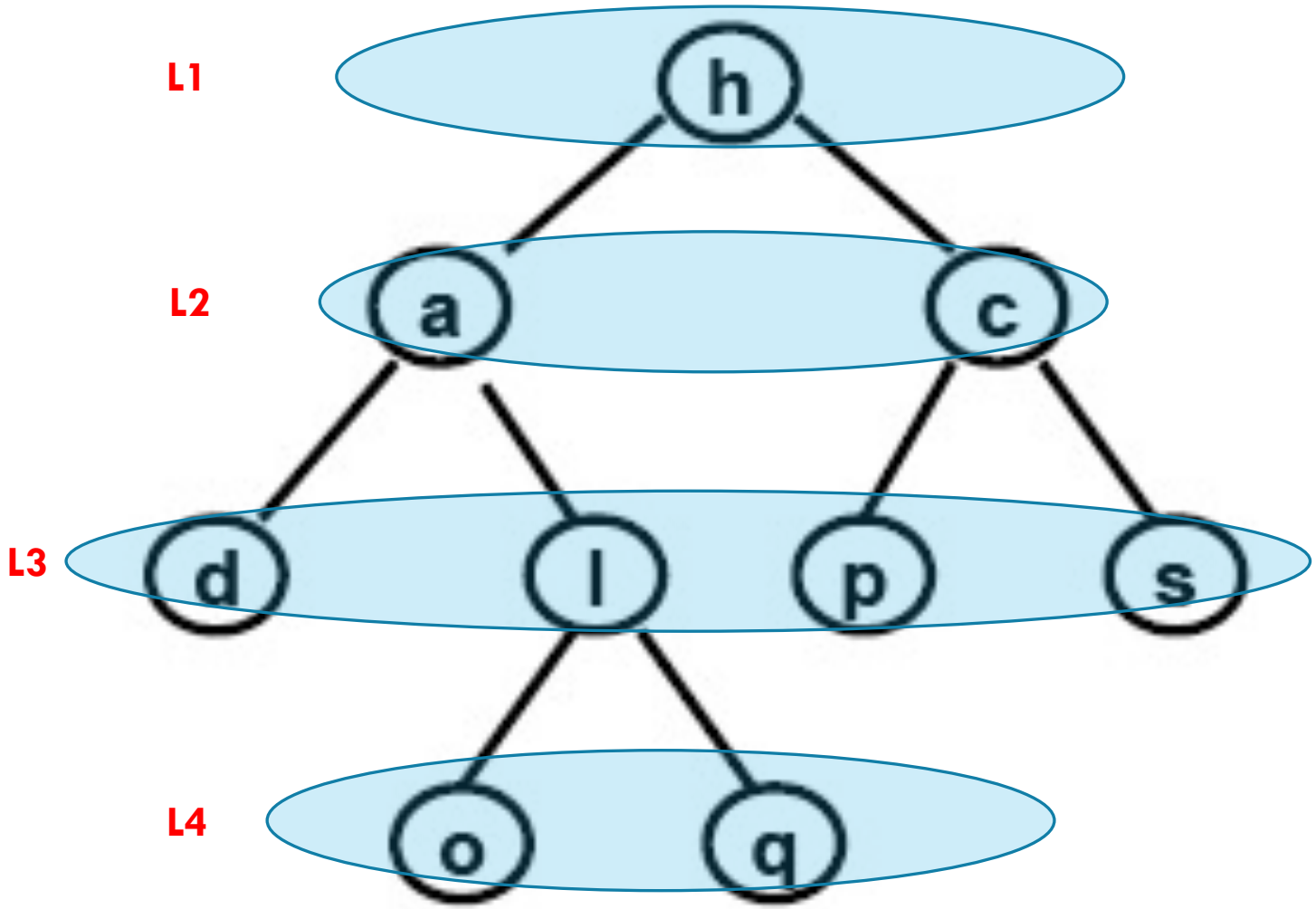
ESERCIZIO 1
PARTE 2

```
89   int numNodi(BTree bt){
90       int count = 1;
91       if (isEmptyTree(bt))
92           return 0;
93       else{
94           if(!isEmptyTree(bt->left))
95               count += numNodi(bt->left);
96           if(!isEmptyTree(bt->right))
97               count += numNodi(bt->right);
98           return count;
99       }
100  }
```

ESERCIZIO 2

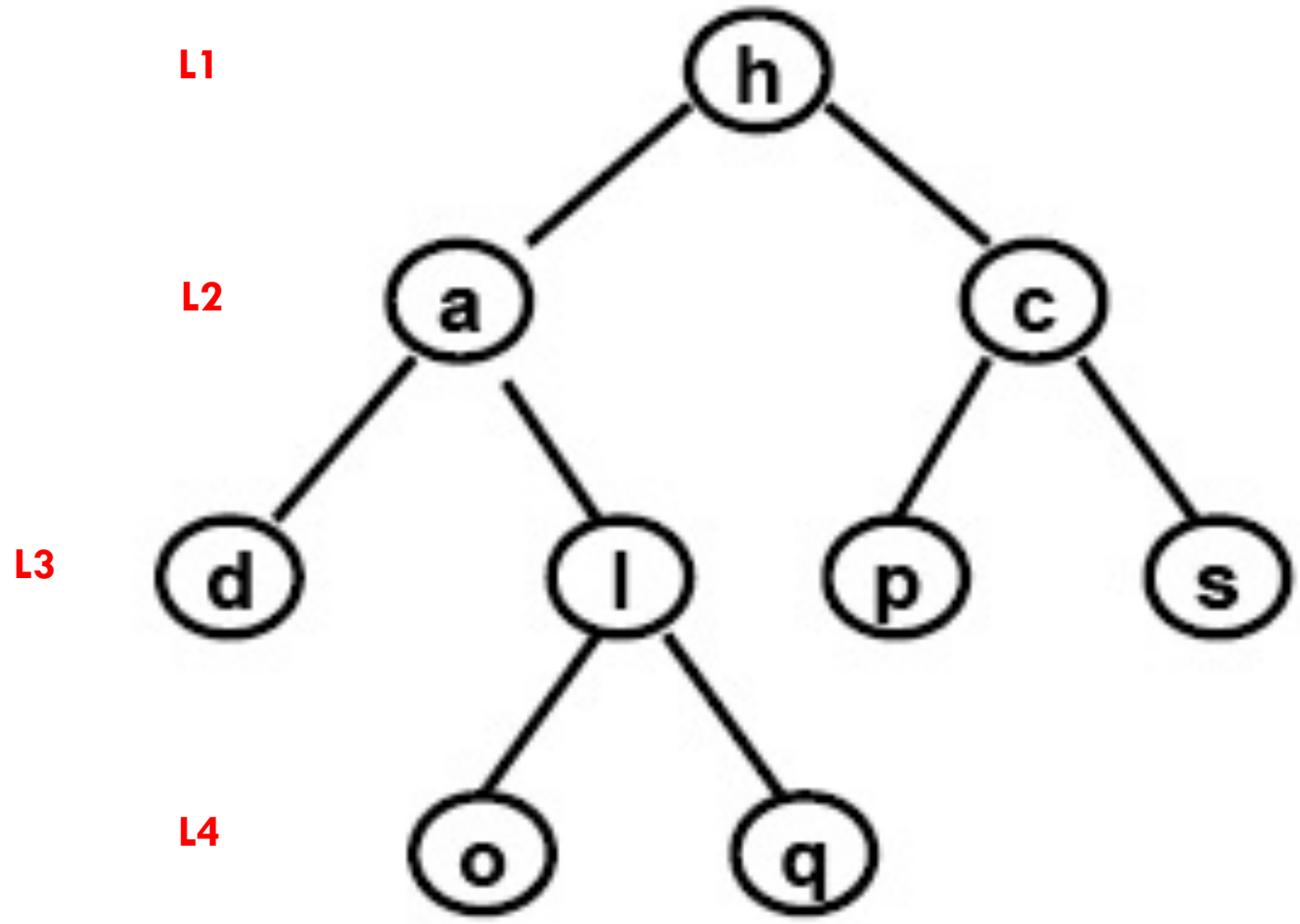
Realizzare una visita per livelli di un albero binario

Suggerimento: usare una coda



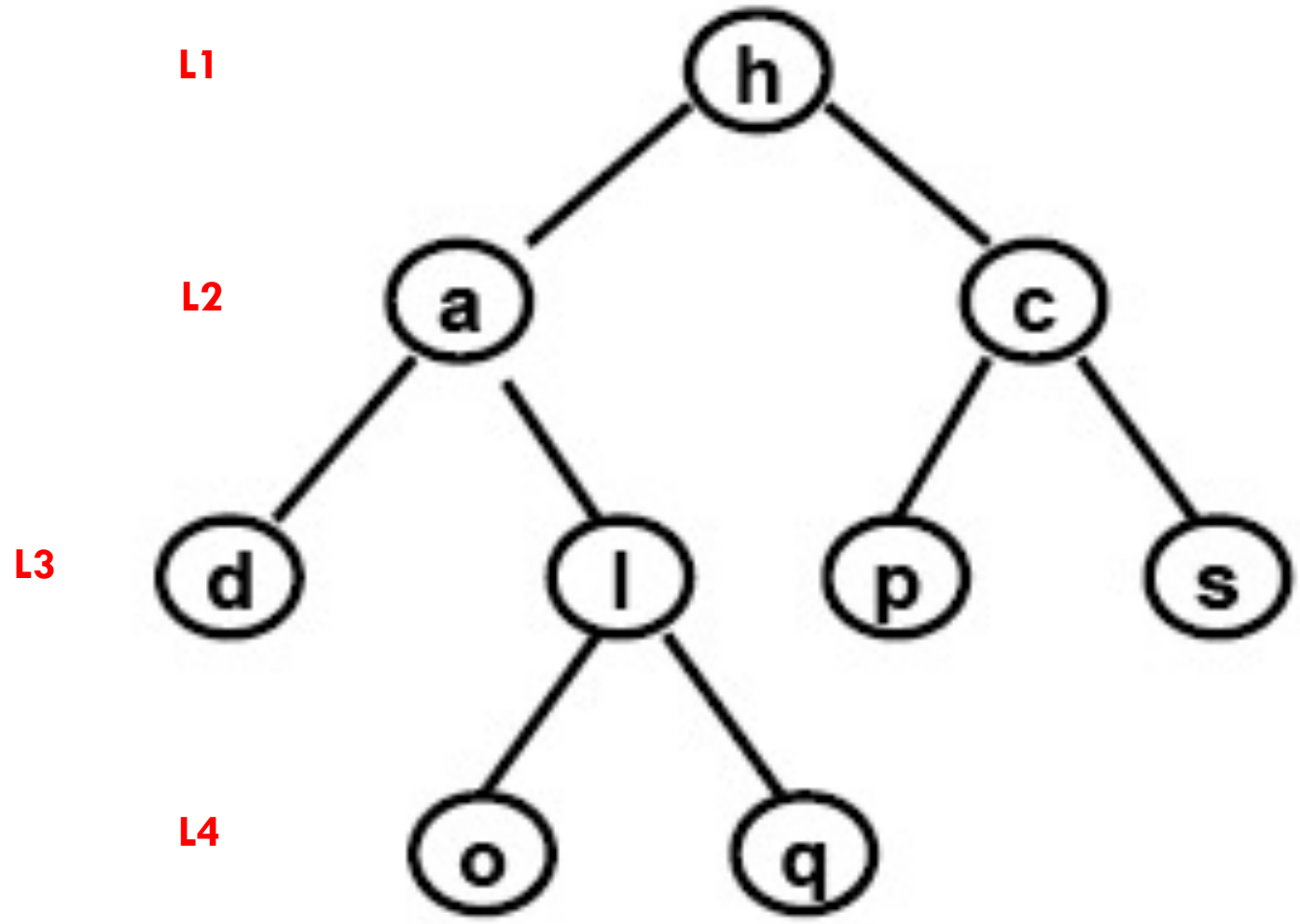
H A C D L P S O Q

H					



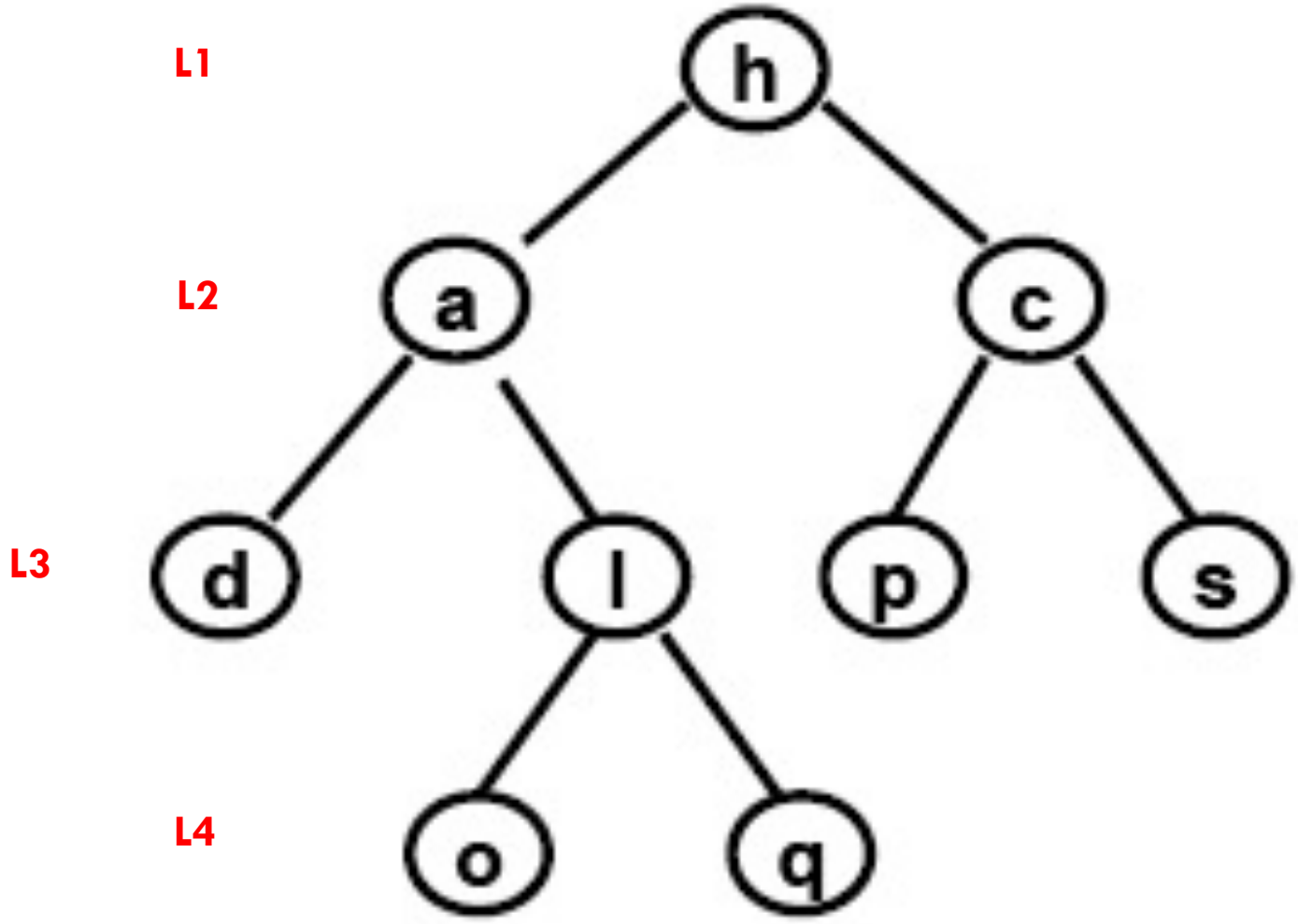
H

C	A				



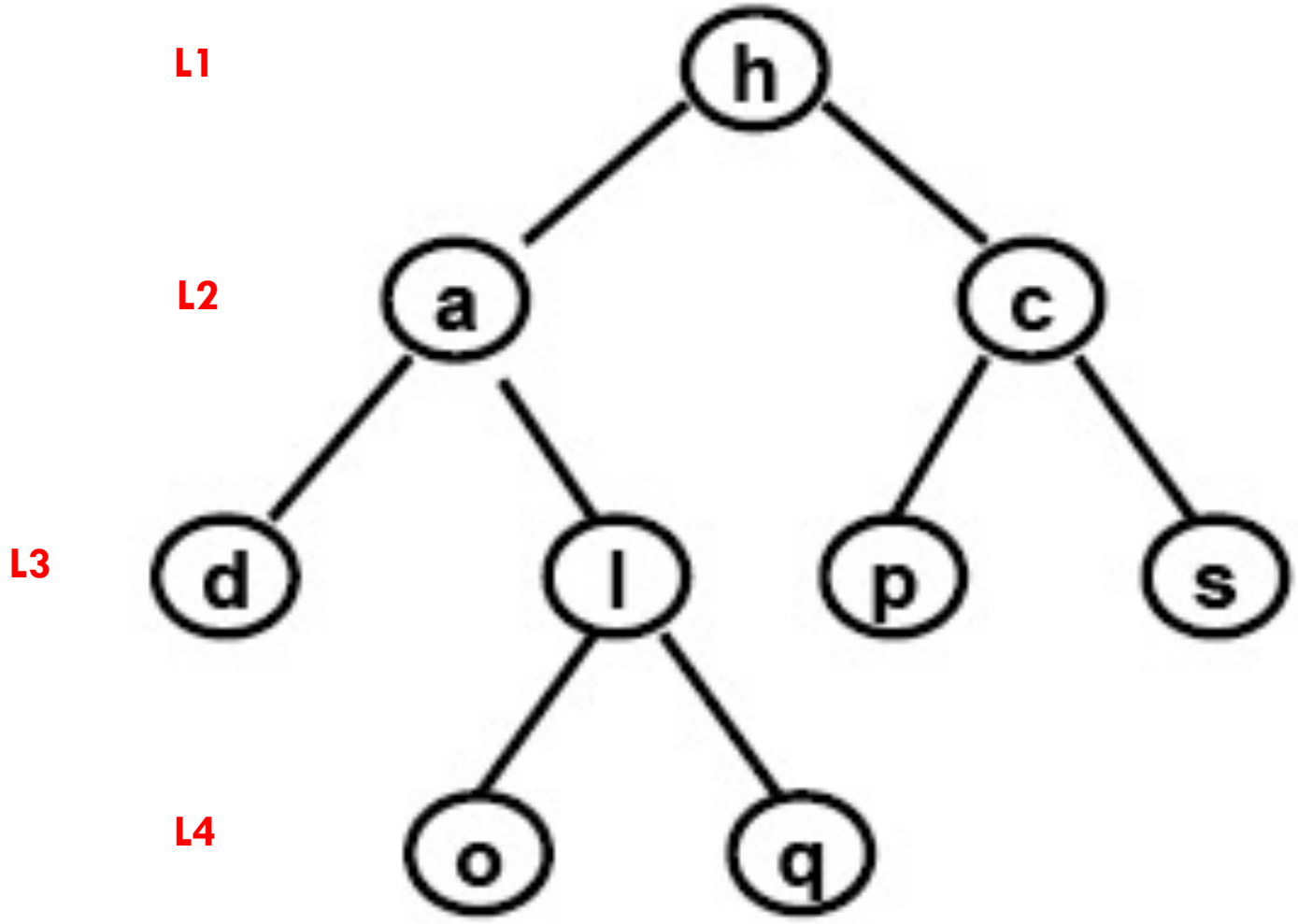
H A

L	D	C			



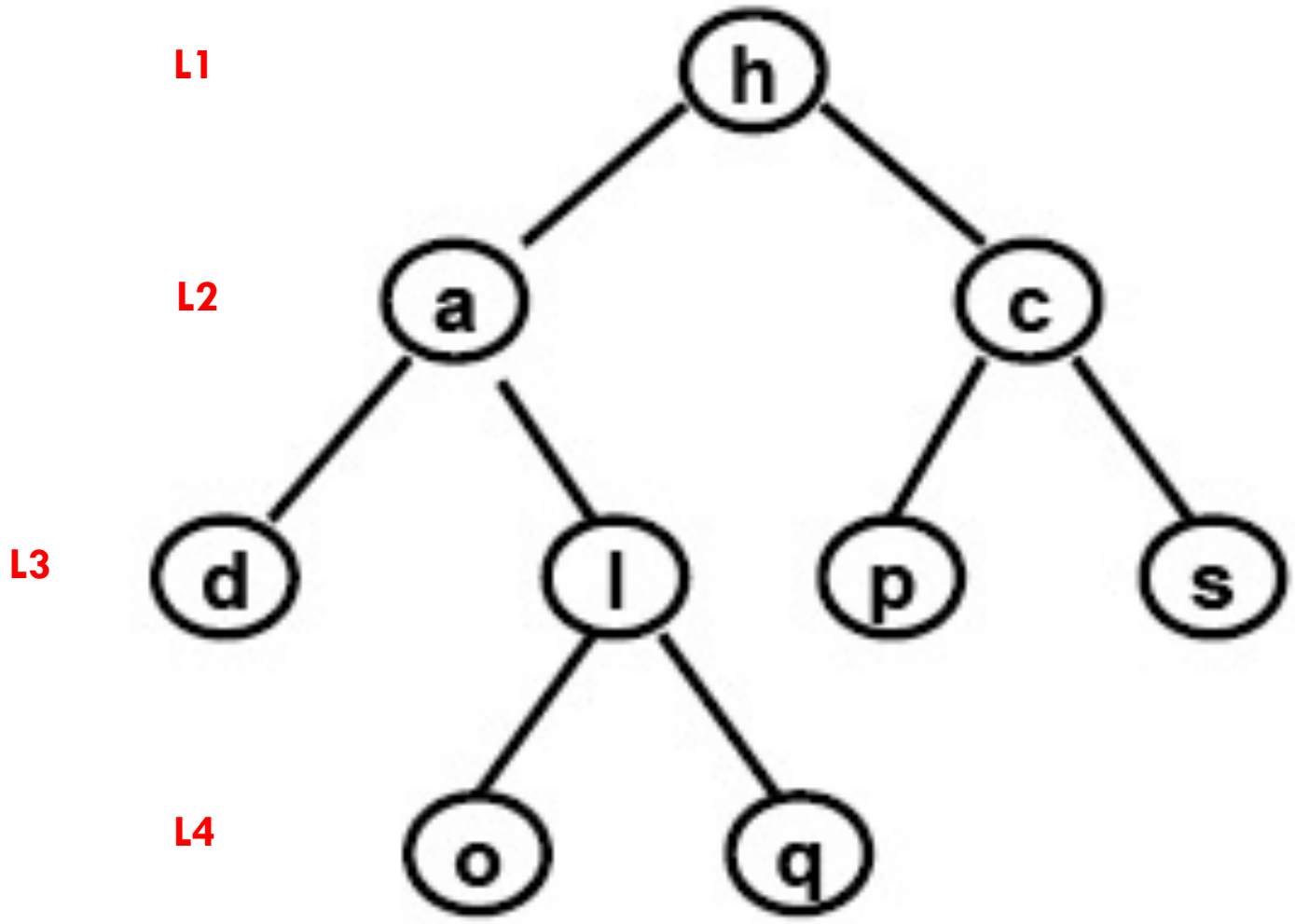
H A C

S	P	L	D		



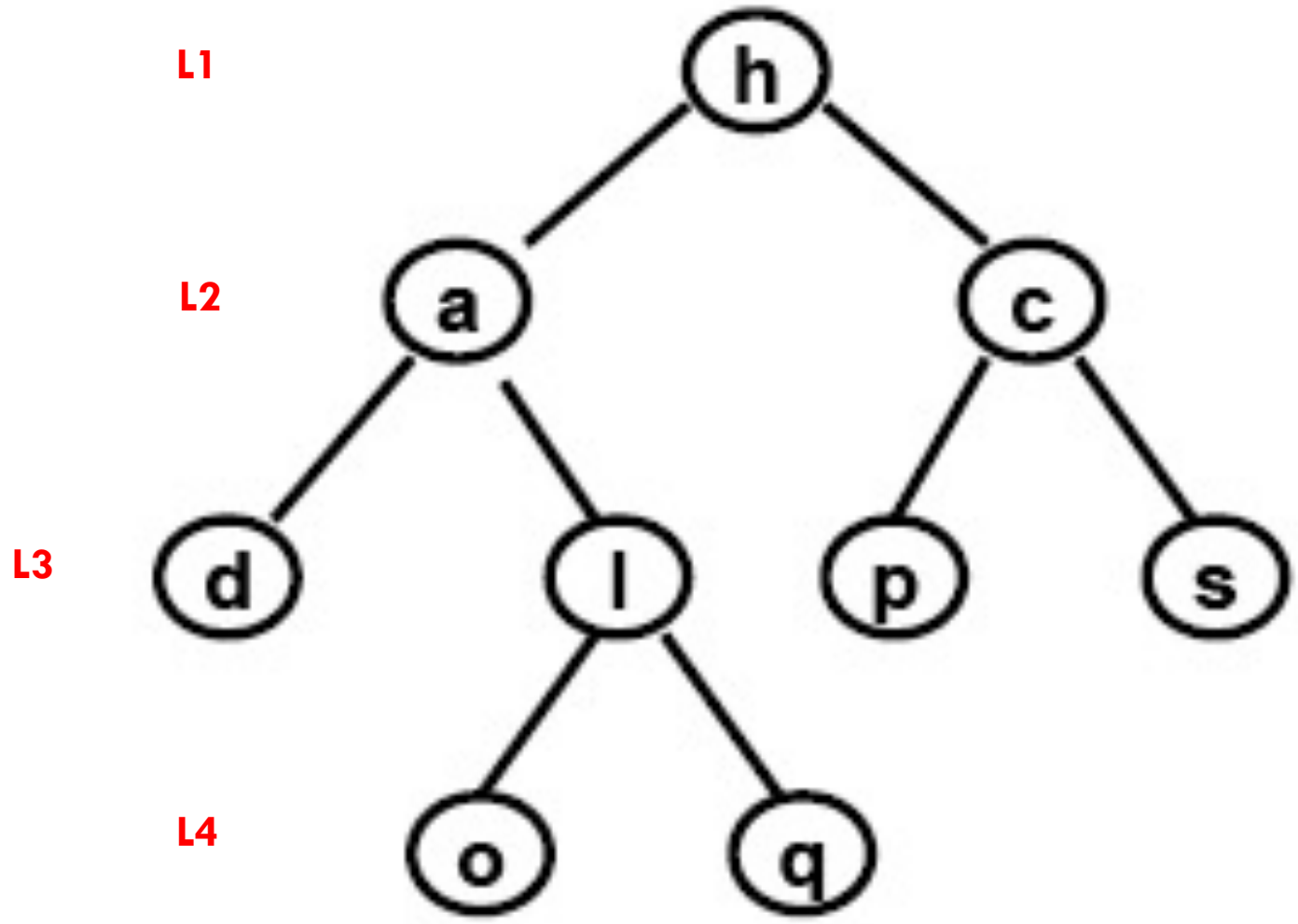
H A C D

S	P	L			



H A C D L

Q	O	S	P		

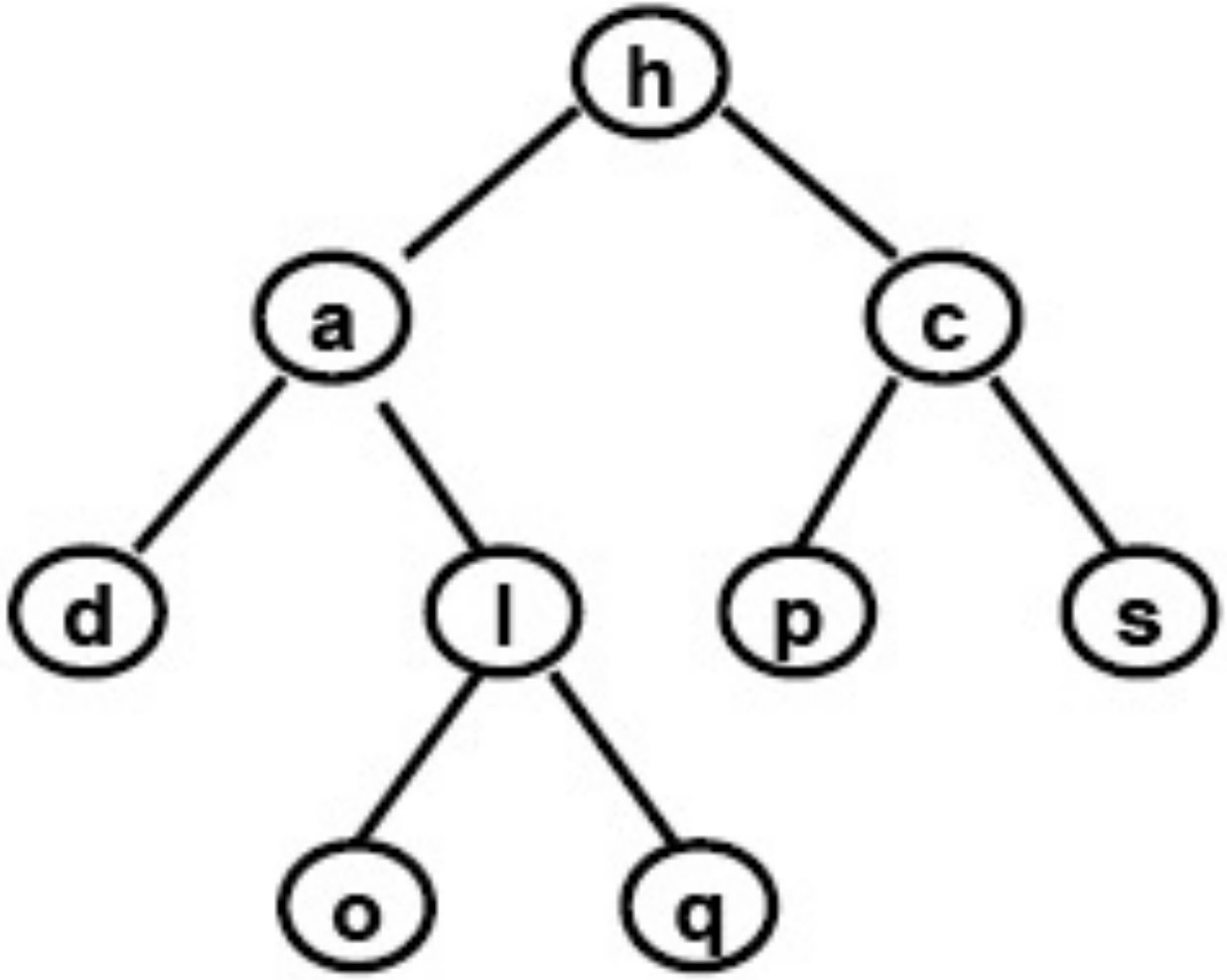


ESERCIZIO 2

```
117  void byLevel(BTree bt){
118      BTree left, right;
119      Queue q = newQueue();
120      enqueue(q, bt);
121      while (!isEmptyQueue(q)){
122          BTree node = dequeue(q);
123          outputItem(node->value);
124          if ((left = getLeft(node))!=NULL)
125              enqueue(q, left);
126          if ((right = getRight(node))!=NULL)
127              enqueue(q, right);
128      }
129  }
```

ESERCIZIO 3

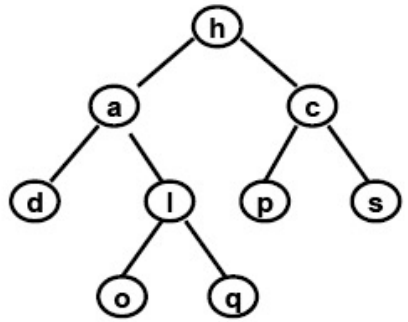
Realizzare le tre visite dell'albero binario in maniera **iterativa,** con l'uso di uno **stack**



- **pre-ordine:**
  1. l'analisi della radice dell'albero
  2. la visita dei due sottoalberi, prima il sinistro, poi il destro
- **post-ordine:**
  1. la visita dei sottoalberi, prima il sinistro e poi il destro
  2. l'analisi della radice dell'albero
- **simmetrica:**
  1. la visita del sottoalbero sinistro
  2. l'analisi della radice
  3. la visita del sottoalbero destro

ESEMPIO:

SIA UN ALBERO BINARIO CHE HA DEI CARATTERI NEI NODI



LA VISITA IN PREORDINE: h a d l o q c p s

LA VISITA IN POSTORDINE: d o q l a p s c h

LA VISITA SIMMETRICA: d a o l q h p c s

29

ESERCIZIO 3