

ESAME DI ALGORITMI E STRUTTURE DI DATI I  
Martedì 10 Febbraio 2004

NOME:  
COGNOME:  
MATRICOLA:

Scrivere in forma leggibile il proprio nome, cognome e matricola sul testo del compito e su ogni foglio consegnato;

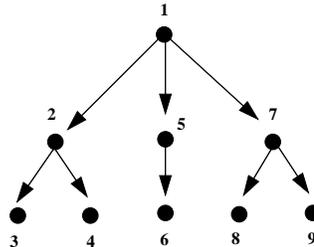
Consegnare solo la bella copia e il testo del compito;

Non è possibile consultare alcun tipo di materiale didattico;

Non è possibile uscire dopo l'inizio dello scritto.

**Esercizio 1** (Punti 30)

Un albero generico è la generalizzazione di un albero binario in cui ogni nodo possiede un numero arbitrario di figli. I figli di ogni nodo sono ordinati da sinistra a destra. L'ordinamento anticipato dei nodi di un albero generico si ottiene ordinando prima la radice e poi ordinando ricorsivamente i nodi di tutti gli alberi radicati sui suoi figli da sinistra a destra. Si veda l'esempio che segue.



Si definisca una opportuna struttura di dati per rappresentare un albero generico e si scriva una procedura *PreorderVisit(x)* che stampa in ordine anticipato le chiavi dei nodi dell'albero generico radicato in *x*.

**Soluzione**

Rappresento ogni nodo *x* dell'albero con un oggetto dotato dei seguenti campi: *c[x]* che punta al primo figlio di *x*, oppure è NIL se *x* non ha figli, *r[x]* che punta al fratello destro di *x*, oppure è NIL se *x* non ha fratelli destri, e *key[x]* che contiene la chiave di *x*.

Propongo una soluzione iterattiva che utilizza una pila ausiliaria  $S$ .

---

**Algoritmo 1** *PreorderVisit(x)*

---

*PreorderVisit(x)*

```
1:  $S \leftarrow \emptyset$ 
2: while  $x \neq \text{NIL}$  do
3:   print  $\text{key}[x]$ 
4:    $\text{Push}(S, x)$ 
5:    $x \leftarrow c[x]$ 
6: end while
7: while not  $\text{StackEmpty}(S)$  do
8:    $x \leftarrow r[\text{Pop}(S)]$ 
9:   while  $x \neq \text{NIL}$  do
10:    print  $\text{key}[x]$ 
11:     $\text{Push}(S, x)$ 
12:     $x \leftarrow c[x]$ 
13:   end while
14: end while
```

---

Una versione ricorsiva è la seguente:

---

**Algoritmo 2** *PreorderVisit(x)*

---

*PreorderVisit(x)*

```
1: if  $x \neq \text{NIL}$  then
2:   print  $\text{key}[x]$ 
3:    $x \leftarrow c[x]$ 
4:   while  $x \neq \text{NIL}$  do
5:      $\text{PreorderVisit}(x)$ 
6:      $x \leftarrow r[x]$ 
7:   end while
8: end if
```

---

Si noti che il tempo impiegato da entrambi gli algoritmi è lineare nel numero di nodi, mentre lo spazio, cioè la dimensione della pila, è proporzionale rispetto all'altezza dell'albero.