

Esercizio 1.

Dato un albero binario e i tre tipi di visita in Preordine, Inordine e Postordine, si supponga che siano definite tre funzioni, PRE , IN e $POST$, che restituiscono la posizione relativa di un nodo nell'attraversamento corrispondente. Per esempio, se l'attraversamento Inordine di un albero è ABC e il Preordine è BAC, allora:

$IN(A)=1, IN(B)=2, IN(C)=3$

$PRE(A)=2, PRE(B)=1, PRE(C)=3$

Quesito 1.

Completare la tabella sotto riportata, indicando se le condizioni indicate in colonna, in un generico albero binario, sono vere per tutte le coppie di nodi m e n tali che n sia antenato di m (prima riga) oppure nel caso in cui m sia antenato di n (seconda riga).

Per esempio, si inserisca il valore "VERO" nell'ultima cella della seconda riga se si ritiene che se m è un antenato di n allora è vero che n precede m in un attraversamento in Postordine, o il valore FALSO altrimenti.

	$PRE(n) < PRE(m)$	$IN(n) < IN(m)$	$POST(n) < POST(m)$
n è antenato di m			
m è antenato di n			

N.B.: un nodo A è un *antenato* di un nodo B se nel cammino dalla radice dell'albero a B si passa attraverso A . Gli antenati sono quindi padri, nonni, ecc.

SOLUZIONE

	$PRE(n) < PRE(m)$	$IN(n) < IN(m)$	$POST(n) < POST(m)$
n è antenato di m	VERO	FALSO	FALSO
m è antenato di n	FALSO	FALSO	VERO

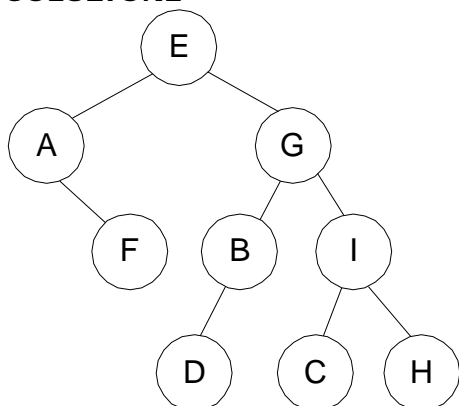
Quesito 2.

Si tracci l'albero binario per il quale sono definiti i seguenti attraversamenti:

Preordine: E A F G B D I C H

Inordine: A F E D B G C I H

SOLUZIONE



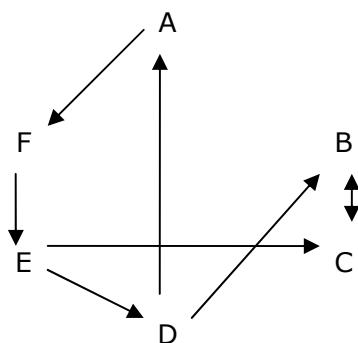
Esercizio 2.

Dato il grafo G , rappresentato dalla seguente lista di adiacenza

A -> F
B -> C
C -> B
D -> A -> B
E -> C -> D
F -> E

si risponda ai seguenti quesiti:

1. Disegnare G
2. G è orientato?
3. G è connesso?
4. G è aciclico?
5. Fornire la matrice delle adiacenze di G .

SOLUZIONE

2. G è orientato
3. G è semplicemente connesso, non è fortemente connesso
4. G non è aciclico: contiene il ciclo AFEDA

5. Matrice delle adiacenze

	A	B	C	D	E	F
A	0	0	0	0	0	1
B	0	0	1	0	0	0
C	0	1	0	0	0	0
D	1	1	0	0	0	0
E	0	0	1	1	0	0
F	0	0	0	0	1	0

Esercizio 3

Si tratteggi un algoritmo, basato sul paradigma del *divide et impera*, che, ricevendo come parametri di ingresso due numeri interi positivi a ed n , calcola la potenza a^n effettuando un numero di operazioni aritmetiche $\theta(\log n)$.

Si argomenti nel modo più preciso possibile che la complessità dell'algoritmo proposto è quella richiesta.

SOLUZIONE

L'algoritmo si basa sulla proprietà

$$a^n = \begin{cases} (a^{n/2})^2 & \text{se } n \text{ è pari;} \\ (a^{n/2})^2 \cdot a & \text{se } n \text{ è dispari} \end{cases}$$

```
int potDivEtImp (int a, int n) {  
    int t;  
    if (n==1) return a;  
    t = potDivEtImp(a, n/2);  
    if (n%2 == 1)  
        return a * t * t;  
    else return t * t;  
}
```

Il numero di chiamate ricorsive della funzione è $\theta(\log n)$ perché ad ogni passo di ricorsione il valore dell'esponente viene dimezzato.

