

Esercizio 1

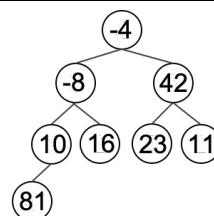
Scrivere una funzione **esercizio1** che avendo in input un albero binario A (dove ogni nodo ha come valore informativo un numero intero) e due interi x e y , restituisca `true` se x e y sono *related nodes* in A , altrimenti `false`. Dati due interi x e y , si dicono *related nodes* nell'albero binario A se x e y sono due nodi di A (quindi esistono due nodi in A che abbiano come valore informativo rispettivamente x e y), e i due nodi sono sullo stesso livello ma hanno padre differente.

L'albero è rappresentato da una classe `AlberoB`. Sia a una istanza della classe, l'interfaccia è la seguente:

- `a.figlio(SIN/DES)` restituisce il figlio sinistro o destro dell'albero a se esiste, altrimenti restituisce la costante `NULLO`;
- `a.padre()` restituisce il padre dell'albero a se esiste, altrimenti restituisce la costante `NULLO`;
- `a.radice()` restituisce il valore informativo associato all'albero a (il valore intero contenuto all'interno del nodo);
- `a.foglia()` restituisce `true` se l'albero a è una foglia, altrimenti `false`.

Si può assumere che nell'albero non vi siano due nodi con lo stesso valore informativo.

Esempio: dati in input 10 e 11, la funzione restituirà `true`, poiché esistono due nodi con valori informativi rispettivamente 10 e 11, e sono allo stesso livello con padre differente; dati in input 10 e 23, la funzione restituirà `true`; dati in input -8 e 42, la funzione restituirà `false`, poiché sono sullo stesso livello ma hanno lo stesso padre (il nodo con valore informativo -4).



ok

Esercizio 2

Scrivere una funzione **esercizio2** che prenda in input un grafo diretto $G(V,E)$, un intero $1 < k \leq |V|$ e un intero positivo s , e restituisca `true` se esiste un sottoinsieme U di V tale che le seguenti condizioni siano soddisfatte:

- U contiene al più k nodi, ovvero $|U| \leq k$,
- per ogni arco (i,j) in E , solo una delle due è vera: i deve essere in U oppure j non deve essere in U ,
- la somma di tutti i nodi in U non può essere uguale a s .

Se tale sottoinsieme non esiste, la funzione restituisce `false`.

Il grafo è rappresentato da una classe `Grafo` con la seguente interfaccia (con g un'istanza della classe):

- `g.n()` restituisce il numero di nodi del grafo,
- `g.m()` restituisce il numero di archi del grafo,
- `g(i,j)` restituisce `true` se esiste l'arco diretto tra il nodo i e il nodo j .

I nodi sono etichettati da 0 a $g.n() - 1$. La somma di un insieme di nodi è data dalla somma delle etichette dei nodi stessi.

Esempio: in questo caso la funzione restituirà `true` poiché è possibile creare un sottoinsieme U che rispetti le condizioni di cui sopra. Una possibile soluzione è data dal sottoinsieme $U = \{1,2\}$ il quale

- contiene al più k nodi,
- per ogni arco (i,j) : o il nodo i appartiene ad U oppure il nodo j non appartiene ad U ,
- la somma dei nodi di U è uguale a 3 che è diversa da s

