

ok **Esercizio 1**

Scrivere una funzione **esercizio1** che prenda in input un grafo non orientato G , dove ogni nodo in G ha associato un valore intero positivo chiamato potenziale. Inoltre, per ogni nodo v si definisce il suo stato come 1 se il potenziale di v è strettamente maggiore della media del potenziale dei suoi nodi direttamente adiacenti, altrimenti lo stato si definisce come 0. La funzione deve ritornare il nodo con stato pari a 1 e con il potenziale più alto.

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

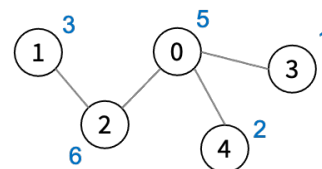
- $g.n()$ restituisce il numero di nodi del grafo,
- $g.m()$ restituisce il numero di archi del grafo,
- $g(i, j)$ restituisce `true` se esiste un arco tra il nodo i e il nodo j .
- $g.pot(i)$ restituisce il potenziale del nodo i .

I nodi sono etichettati da 0 a $g.n() - 1$. Se esistono due nodi con lo stesso potenziale più alto e lo stato pari a 1, restituire il nodo con l'indice minore.

Esempio: in questa immagine, il potenziale di ogni nodo è indicato dal valore in blu vicino il nodo. Ad esempio, il nodo 0 ha potenziale pari a 5. Lo stato di ogni nodo è il seguente:

- nodo 0, potenziale 5, stato 1 (poiché il potenziale, 5, è strettamente maggiore della media del potenziale dei nodi adiacenti $(6+1+2) / 3 = 3$)
- nodo 1, potenziale 3, stato 1
- nodo 2, potenziale 6, stato 1
- nodo 3, potenziale 1, stato 0
- nodo 4, potenziale 2, stato 0

La funzione in questo caso restituisce 2, poiché il nodo 2 è il nodo con stato 1 e potenziale più alto tra tutti i nodi con stato 1.



Esercizio 2 OK

Scrivere una funzione **esercizio2** che prenda in input una collezione V di array, dove ogni array contiene esattamente 4 numeri interi, e un numero intero positivo k , e restituisca `true` se è possibile scegliere almeno k array tale che per ogni coppia di array scelta, tra i due array deve esserci al massimo un solo elemento in comune.

Si assuma:

- V sia rappresentato da un `vector<vector<int>>` (`ArrayList<ArrayList<Integer>>` se si usa Java),
- Ogni vettore in V contiene esattamente 4 numeri interi tutti diversi tra loro.

Esempio: la funzione dovrà restituire `true` poiché è possibile scegliere almeno $k = 3$ array tale che ogni coppia di array scelta ha al massimo un solo elemento in comune.

Una possibile soluzione è data dagli array $[42, 3, 1, 5]$, $[2, 8, 21, 83]$, $[22, 23, 24, 42]$ poiché

- Il primo array non ha elementi in comune con il secondo array, e ne ha solo uno con il terzo array (42),
- Il secondo array non ha elementi in comune con il terzo array.

$V = \{[42, 3, 1, 5], [8, 26, 3, 5], [2, 8, 21, 83], [7, 21, 83, 5], [22, 23, 24, 42]\}$
 $k = 3$