

## 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 latenza. Inoltre, per ogni nodo  $v$  si definisce il suo stato come 1 se la latenza di  $v$  è strettamente minore della media della latenza 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 la latenza più alta.

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

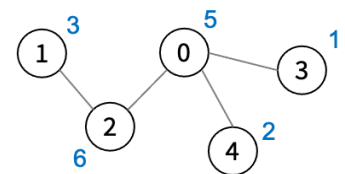
- $g.n()$  restituisce il numero di nodi del grafo,
- $g.m()$  restituisce il numero di archi del grafo,
- $g(i, j)$  restituisce `true` se esiste un arco tra il nodo  $i$  e il nodo  $j$ .
- $g.lat(i)$  restituisce la latenza del nodo  $i$ .

I nodi sono etichettati da 0 a  $g.n() - 1$ . Se esistono due nodi con la stessa latenza più alta e lo stato pari a 1, restituire il nodo con l'indice minore. Se non esiste un nodo con stato 1, restituire -1.

*Esempio:* in questa immagine, la latenza di ogni nodo è indicata dal valore in blu vicino al nodo. Ad esempio, il nodo 0 ha latenza pari a 5. Lo stato di ogni nodo è il seguente:

- nodo 0, latenza 5, stato 0 (poiché la latenza, pari a 5, non è minore della media della latenza dei nodi adiacenti  $(6+1+2) / 3 = 3$ )
- nodo 1, latenza 3, stato 1
- nodo 2, latenza 6, stato 0
- nodo 3, latenza 1, stato 1
- nodo 4, latenza 2, stato 1

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



```

int esercizio1(const Grafo& G)
{
    int nodomax=-1, latenzamax=-1;

    for (int i=0; i<G.n(); i++){
        float latenza=0;
        int count=0;
        for (int j=0; j<G.n(); j++){
            if (i!=j && G(i,j)) {
                latenza+=G.lat(j);
                count++;
            }
        }
        latenza=latenza/count; // calcolo la media
        if ((G.lat(i)<latenza)){ //stato 1
            if(G.lat(i)>latenzamax){
                nodomax=i; latenzamax=G.lat(i);
            }
        }
    }

    return i;
}

```