

## Esercizio

Scrivere una funzione **esercizio1** che prenda in input un grafo orientato e pesato  $G$ , dove ogni nodo in  $G$  ha associato un valore intero positivo chiamato deposito. La funzione deve ritornare la stringa YES se per ogni nodo  $v$  del grafo valgono le seguenti due condizioni:

- La somma dei pesi degli archi entranti in  $v$  è minore del suo deposito, e
- Il deposito di ogni nodo avente un arco entrante in  $v$  è maggiore o uguale al deposito di  $v$ .

Se un nodo non ha archi entranti, allora le due condizioni si assumono vere.

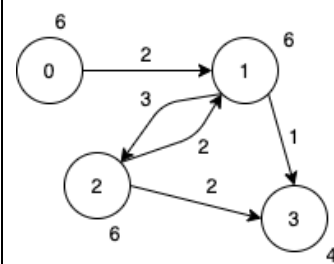
Il grafo è rappresentato da una classe GrafoPesato con la seguente interfaccia (con  $g$  un'istanza della classe):

- $g.n()$  restituisce il numero di nodi del grafo,
- $g.m()$  restituisce il numero di archi del grafo,
- $g(i,j)$  restituisce il peso dell'arco tra il nodo  $i$  e il nodo  $j$ , altrimenti 0 (l'arco non esiste),
- $g.dep(i)$  restituisce il deposito del nodo  $i$ .

I nodi sono etichettati da 0 a  $g.n()-1$ . Se le due condizioni di cui sopra non sono soddisfatte per ogni nodo, allora la funzione restituisce la stringa NO.

*Esempio:* in questo caso, la funzione dovrebbe restituire YES poiché le condizioni di cui sopra valgono per ogni nodo. Ad esempio:

- il nodo 0 non ha archi entranti, quindi le proprietà si assumono vere;
- il nodo 1 ha deposito pari a 6 e due archi entranti, la cui somma è 4, quindi minore del deposito; inoltre, i nodi dei suoi archi entranti hanno tutti il deposito uguale a 6,



//VERSIONE 1 INTUITIVA MA NON EFFICIENTE

```
string esercizio(const GrafoPesato& G)
{
    bool verifica=true;
    for (int i=0; i<g.n() && verifica; i++)
    {
        int somma=0;
        for (int j=0;i<g.n() && verifica; j++)
        {
            if ((i!=j) && g(j,i)!=0) somma+=g(j,i);
        }
        if (somma>=g.dep(i)) verifica=false;
    }

    if (!verifica) return "NO";
    else{
        for (int i=0; i<g.n() && verifica; i++)
        {
            for (int j=0;i<g.n() && verifica; j++)
            {
                if ((i!=j) && g(j,i)!=0 && g.dep(j) <= g.dep(i))
                    verifica=false;
            }
        }

        if (!verifica) return "NO";
        else return "YES";
    }
}
```

/// =====

// VERSIONE 2

```
string esercizio(const GrafoPesato& G)
{
    bool verifica=true;
    for (int i=0; i<g.n() && verifica; i++)
    {
        int somma=0;
        for (int j=0;i<g.n() && verifica; j++)
        {
            if ((i!=j) && g(j,i)!=0)
            {
                somma+=g(j,i);
                if (g.dep(j) <= g.dep(i)) verifica=false;
            }
        }
        if (somma>=g.dep(i)) verifica=false;
    }

    if (!verifica) return "NO";
    else return "YES";
}
```