

---

**Graf dirigit amb llistes d'adjacència. Hi ha camí d'un vèrtex a un altre?**

---

**X89807\_ca**

---

Donada la classe *graf* que permet gestionar grafs dirigits i no etiquetats amb  $n$  vèrtexs (els vèrtexs són enters dins l'interval  $[0, n - 1]$ ), cal implementar amb un alhorsime **recursiu** el mètode

```
bool hi_ha_cami(nat ini, nat fi) const;
// Pre: ini i fi són vèrtexs del graf (són menors que n)
// Post: Retorna un booleà indicant si hi ha camí per anar d'ini a fi
```

Les arestes es guarden en llistes d'adjacència: un vector de  $n$  elements que conté vectors amb els successors de cadascun dels  $n$  vèrtexs. Per exemple, un dels jocs de prova públics llegeix aquest graf que conté 5 vèrtexs (mira el PDF de l'enunciat):

les seves arestes estarien guardades en un vector amb 5 llistes d'adjacència, els successors de cadascun dels 5 vèrtexs:

```
0 [2, 1]
1 [3]
2 [1, 4, 3]
3 []
4 [0]
```

Cal enviar a jutge.org la següent especificació de la classe *graf* i la implementació del mètode dins del mateix fitxer (la resta de mètodes públics ja estan implementats). Indica dins d'un comentari a la capçalera del mètode el seu cost en funció del nombre de vèrtexs  $n$  i el nombre d'arestes  $m$  del graf.

```
#include <vector>
using namespace std;
typedef unsigned int nat;
```

```
class graf {
// Graf dirigit i no etiquetat.
// Les arestes es guarden en llistes d'adjacència (vector amb els successors).
public:
// Constructora per defecte. Crea un graf buit.
graf();

// Destructora
~graf();

// Llegeix les dades del graf del canal d'entrada
void llegeix();

bool hi_ha_cami(nat ini, nat fi) const;
// Pre: ini i fi són vèrtexs del graf (són menors que n)
// Post: Retorna un booleà indicant si hi ha camí per anar d'ini a fi
```

```

private:
    nat n; // Nombre de vèrtexs
    nat m; // Nombre d'arestes
    vector<vector<nat> > a; // Vectors amb els successors de cada vèrtex

    // Aquí va l'especificació dels mètodes privats addicionals
};

// Aquí va la implementació del mètode públic hi_ha_cami i privats addicionals

```

Degut a que jutge.org només permet l'enviament d'un fitxer amb la solució del problema, en el mateix fitxer hi ha d'haver l'especificació de la classe i la implementació del mètode *hi\_ha\_cami* (el que normalment estarien separats en els fitxers *.hpp* i *.cpp*). Per testejar la classe disposes d'un programa principal que llegeix un graf i després llegeix diverses parelles de vèrtexs per esbrinar si hi ha camí per anar d'un vèrtex a l'altre.

## Entrada

L'entrada conté un graf: el nombre de vèrtexs, el nombre d'arestes i una llista d'arestes. Cada aresta s'indica pels dos vèrtexs que relaciona. A continuació hi ha diverses parelles de vèrtexs dels quals haurem d'esbrinar si hi ha camí per anar d'un a l'altre.

## Sortida

Per a cada parella de vèrtexs de l'entrada, per exemple  $v1$  i  $v2$ , escriu una línia amb el text "SI hi ha camí de  $v1$  a  $v2$ " o "NO hi ha camí de  $v1$  a  $v2$ ".

## Observació

Només cal enviar la classe requerida i la implementació del mètode *hi\_ha\_cami*. Pots ampliar la classe amb mètodes privats. Segueix estrictament la definició de la classe de l'enunciat. La solució a aquest problema ha de ser **recursiva**. Indica dins d'un comentari a la capçalera del mètode el seu cost en funció del nombre de vèrtexs  $n$  i el nombre d'arestes  $m$  del graf.

### Exemple d'entrada 1

```

1
0

0 0

```

### Exemple d'entrada 2

```

2
0

0 1
1 0

```

### Exemple d'entrada 3

```

2

```

### Exemple de sortida 1

```

SI hi ha camí de 0 a 0

```

### Exemple de sortida 2

```

NO hi ha camí de 0 a 1
NO hi ha camí de 1 a 0

```

```

1
0 1

```

0 1  
1 0

#### Exemple d'entrada 4

2  
2  
0 1  
1 0

0 1  
1 0

#### Exemple d'entrada 5

3  
3  
0 2  
0 1  
1 2

0 1  
0 2  
1 0  
1 2  
2 0  
2 1

#### Exemple d'entrada 6

5  
7  
4 0  
0 2  
0 1  
2 1  
2 4  
2 3  
1 3

0 1  
0 2  
0 3  
0 4  
1 0  
1 2  
1 3  
1 4  
2 0  
2 1  
2 3  
2 4  
3 0  
3 1  
3 2  
3 4  
4 0  
4 1  
4 2  
4 3

#### Exemple de sortida 3

SI hi ha camí de 0 a 1  
NO hi ha camí de 1 a 0

#### Exemple de sortida 4

SI hi ha camí de 0 a 1  
SI hi ha camí de 1 a 0

#### Exemple de sortida 5

SI hi ha camí de 0 a 1  
SI hi ha camí de 0 a 2  
NO hi ha camí de 1 a 0  
SI hi ha camí de 1 a 2  
NO hi ha camí de 2 a 0  
NO hi ha camí de 2 a 1

#### Exemple de sortida 6

SI hi ha camí de 0 a 1  
SI hi ha camí de 0 a 2  
SI hi ha camí de 0 a 3  
SI hi ha camí de 0 a 4  
NO hi ha camí de 1 a 0  
NO hi ha camí de 1 a 2  
SI hi ha camí de 1 a 3  
NO hi ha camí de 1 a 4  
SI hi ha camí de 2 a 0  
SI hi ha camí de 2 a 1  
SI hi ha camí de 2 a 3  
SI hi ha camí de 2 a 4  
NO hi ha camí de 3 a 0  
NO hi ha camí de 3 a 1  
NO hi ha camí de 3 a 2  
NO hi ha camí de 3 a 4  
SI hi ha camí de 4 a 0  
SI hi ha camí de 4 a 1  
SI hi ha camí de 4 a 2  
SI hi ha camí de 4 a 3

### Exemple d'entrada 7

```
6
7
1 5
1 0
3 1
4 0
0 5
5 1
2 3

0 1
0 2
0 3
0 4
0 5
1 0
1 2
1 3
1 4
1 5
2 0
2 1
2 3
2 4
2 5
3 0
3 1
3 2
3 4
3 5
4 0
4 1
4 2
4 3
4 5
5 0
5 1
5 2
5 3
5 4
```

### Exemple de sortida 7

```
SI hi ha camí de 0 a 1
NO hi ha camí de 0 a 2
NO hi ha camí de 0 a 3
NO hi ha camí de 0 a 4
SI hi ha camí de 0 a 5
SI hi ha camí de 1 a 0
NO hi ha camí de 1 a 2
NO hi ha camí de 1 a 3
NO hi ha camí de 1 a 4
SI hi ha camí de 1 a 5
SI hi ha camí de 2 a 0
SI hi ha camí de 2 a 1
SI hi ha camí de 2 a 3
NO hi ha camí de 2 a 4
SI hi ha camí de 2 a 5
SI hi ha camí de 3 a 0
SI hi ha camí de 3 a 1
NO hi ha camí de 3 a 2
NO hi ha camí de 3 a 4
SI hi ha camí de 3 a 5
SI hi ha camí de 4 a 0
SI hi ha camí de 4 a 1
NO hi ha camí de 4 a 2
NO hi ha camí de 4 a 3
SI hi ha camí de 4 a 5
SI hi ha camí de 5 a 0
SI hi ha camí de 5 a 1
NO hi ha camí de 5 a 2
NO hi ha camí de 5 a 3
NO hi ha camí de 5 a 4
```

### Informació del problema

Autoria: Jordi Esteve

Generació: 2026-01-25T17:07:29.334Z

© Jutge.org, 2006–2026.

<https://jutge.org>

