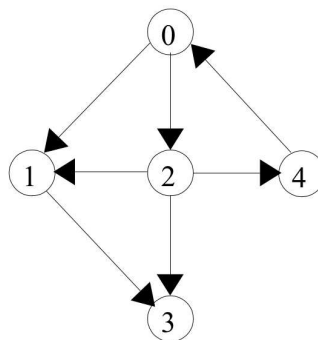


Graf dirigit amb llistes d'adjacència. Successors diferents dels successors de cada vèrtex X53729_ca

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 el mètode

```
vector<nat> quants_succ_succ() const;
// Pre: Cert
// Post: Retorna quants successors diferents tenen els successors de cada vèrtex del graf
```

Les arestes es guarden en llistes d'adjacència: un vector de *n* elements que conté vectors amb els successors de cadascun dels *n* vèrtexs. Un dels jocs de prova públics és 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]
```

el qual donaria com a resultat el vector 3 0 2 0 2, indicant que hi ha 3 vèrtexs diferents com a successors dels successors del vèrtex 0 (el 1, el 3 i el 4), hi ha 0 del vèrtex 1, hi ha 2 del vèrtex 2 (el 0 i el 3), hi ha 0 del vèrtex 3 i hi ha 2 del vèrtex 4 (el 1 i el 2).

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 ();

    vector <nat> quants_succ_succ() const;
    // Pre: Cert
    // Post: Retorna quants successors diferents tenen els successors de cada vèrtex
    // del graf

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 quants_succ_succ 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 `quants_succ_succ` (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 crida el mètode `quants_succ_succ`.

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.

Sortida

Escriu una línia amb el nombre de successors diferents que tenen els successors de cada vèrtex del graf separats per espais.

Observació

Només cal enviar la classe requerida i la implementació del mètode `quants_succ_succ`. Pots ampliar la classe amb mètodes privats. Segueix estrictament la definició de la classe de l'enunciat.

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

Exemple de sortida 1

0

Exemple d'entrada 2

2
0

Exemple de sortida 2

0 0

Exemple d'entrada 3

2
1
0 1

Exemple de sortida 3

0 0

Exemple d'entrada 4

2
2
0 1
1 0

Exemple de sortida 4

1 1

Exemple d'entrada 5

3
4
0 2
0 1
1 2
2 0

Exemple de sortida 5

2 1 2

Exemple d'entrada 6

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

Exemple de sortida 6

3 0 2 0 2

Exemple d'entrada 7

6
9
1 5
1 0
3 1
4 0
0 5
5 1
2 3
0 1
5 0

Exemple de sortida 7

3 3 1 2 2 3

Informació del problema

Autor : Jordi Esteve

Generació : 2021-06-02 09:57:49

© *Jutge.org*, 2006–2021.
<https://jutge.org>