
Control PRO2 - Torn únic (tardor 2018)

X45910_ca

Heu vist al laboratori diferents versions de la classe `Cjt_estudiants`. En aquest exercici treballarem amb una versió on s'ha afegit una operació `estudiant_nota_max` per obtenir l'estudiant amb millor nota amb la següent especificació:

```
Estudiant estudiant_nota_max() const;
/* Pre: el paràmetre implícit conté almenys un estudiant amb nota */
/* Post: el resultat és l'estudiant del paràmetre implícit amb nota
màxima; si en té més d'un, és el de DNI més petit */
```

Per fer eficient aquesta operació hi ha un atribut a la classe, anomenat `imax`, que val -1 si cap estudiant té nota o bé indica la posició on es troba l'estudiant amb millor nota. En cas que més d'un estudiant tingui la millor nota, `imax` indica la posició de l'estudiant amb millor nota amb `dni` més petit. El valor d'aquest camp s'ha de gestionar a tota operació modificadora de la classe `Cjt_estudiants`.

També apareixen les operacions `afegir_estudiant` i `modificar_estudiant`, amb les següents especificacions:

```
void afegir_estudiant(const Estudiant &est, bool& b);
/* Pre: el paràmetre implícit no està ple */
/* Post: b = indica si el p.i. original conté un estudiant amb el dni d'est;
si b = fals, s'ha afegit l'estudiant est al paràmetre implícit */

void modificar_estudiant(const Estudiant &est, bool& b);
/* Pre: cert */
/* Post: b indica si el paràmetre implícit original tenia un estudiant
amb el dni d'est; si b, aquest ha quedat substituït per est */
```

Hem decidit estendre aquesta classe amb dues **noves** funcionalitats que haureu d'implementar:

```
void Cjt_estudiants::unir_conjunts(const Cjt_estudiants & c);
/* Pre: tots els dni de c hi són al p.i. */
/* Post: cada Estudiant del paràmetre implícit conté la millor nota entre l'original
i la que pugui tenir a c */

void Cjt_estudiants::actualitzar_conjunt();
/* Pre: a l'entrada estàndar hi ha una seqüència d'Estudiant (seguida d' un Estudiant
amb dni 0 que no forma part de la seqüència), amb els dni ordenats creixentment
i tots hi són al p.i. */
/* Post: cada Estudiant del paràmetre implícit conté la millor nota entre l'original
i la que pugui tenir a la seqüència */
```

Com sempre, a efectes d'aquest exercici es considera que tenir qualsevol nota vàlida és millor que no tenir nota.

Observació

A *public files* (icona del gatet) trobareu el fitxer `solucio.cc`. En aquest fitxer haureu de fer una implementació eficient de les operacions `unir_conjunts` i `actualitzar_conjunt`. El vostre `solucio.cc` no pot contenir la implementació d'altres operacions de la classe però sí que podeu definir operacions que no pertanyin a cap classe. L'únic fitxer que cal lliurar es `solucio.cc`.

També a *public files* us proveïm amb material addicional comprimit en un fitxer `.tar`. Podeu descomprimir aquest fitxer amb la comanda

```
tar -xvf nom_fitxer.tar
```

Aquest material addicional consisteix en els següents fitxers:

- `Cjt_estudiants.hh`: les capçaleres i l'especificació Pre/Post de totes les operacions públiques i privades d'aquesta versió de la classe `Cjt_estudiants`, així com la definició dels camps privats. És molt important que la implementació de les operacions que us hem encarregat tingui en compte i preservi l'invariant de la representació de la classe `Cjt_estudiants`
- `Cjt_estudiants.cc`: la implementació de totes les operacions de la nova versió de la classe `Cjt_estudiants` tret de les operacions `unir_conjunts` i `actualitzar_conjunt`
- `Estudiant.hh`: les capçaleres i l'especificació de la classe `Estudiant` i la definició dels seus atributs
- `Estudiant.cc`: la implementació dels mètodes de la classe `Estudiant`
- `pro2.cc`: un programa principal que podeu fer servir per provar les operacions públiques d'aquesta versió de la classe `Cjt_estudiants`; està garantit que els jocs de proves del jutge compleixen les precondicions de totes les crides que s'hi fan
- `llegeixme.txt`: instruccions per a generar l'executable del programa `pro2` i provar-lo
- `sample.inp`: entrada del joc de proves públic
- `sample.cor`: sortida del joc de proves públic
- `solucio.cc`: explicat previament

És fonamental que la solució sigui **eficient en temps i espai**. En particular, s'han d'evitar instruccions innecessàries (especialment bucles o crides a operacions costoses) i no es poden fer servir objectes auxiliars de les classes `vector` o `Cjt_estudiants`. No es pot emprar cap estructura de dades que no hagi aparegut a les sessions 1-4 de laboratori.

Quan feu els enviaments el Jutge us indicarà quants jocs de proves passeu i de quin tipus (públic o privat). El joc de proves anomenat `sample` es mostra a sota. El `privat1` és molt semblant al `sample`. Els `privat2` i `privat3` proven situacions especials.

Exemple d'entrada 1

5	222 3.33
111 1.11	444 5.17
222 2.22	0 0
333 545	-5
444 -8	-6
555 5.32	-4 444 8.88
-2	-1 666 4.44
3	-5
111 1.77	-6
333 2.21	-4 444 3.22
444 4.98	-5
-5	-6
-6	-1 200 2.45
-3	-1 777 8.65
	-1 300 4.99
	-1 400 -10

-5
-6
-2
10
111 2.13
200 -30
222 2.20
300 5.03
333 2.21
400 1.05
444 4.43
555 6.32
666 14
777 8.99
-5
-6
-3
111 1.11
222 -20
300 5.03
444 9.99
555 6.32
666 4.45
0 8
-5
-6
-7

Exemple de sortida 1

Conjunt:
111 1.77
222 2.22
333 2.21
444 4.98
555 5.32

Nota max: 555 5.32

Conjunt:
111 1.77
222 3.33
333 2.21
444 5.17
555 5.32

Nota max: 555 5.32

Conjunt:
111 1.77
222 3.33
333 2.21
444 8.88
555 5.32
666 4.44

Nota max: 444 8.88

Conjunt:
111 1.77
222 3.33
333 2.21
444 3.22
555 5.32
666 4.44

Nota max: 555 5.32

Conjunt:
111 1.77
200 2.45
222 3.33
300 4.99
333 2.21
400 NP
444 3.22
555 5.32
666 4.44
777 8.65

Nota max: 777 8.65

Conjunt:
111 2.13
200 2.45
222 3.33
300 5.03
333 2.21
400 1.05

444 4.43
555 6.32
666 4.44
777 8.99

Nota max: 777 8.99

Conjunt:
111 2.13
200 2.45
222 3.33

300 5.03
333 2.21
400 1.05
444 9.99
555 6.32
666 4.45
777 8.99

Nota max: 444 9.99

Informació del problema

Autoria: Professors de PRO2

Generació: 2026-01-25T16:21:38.665Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>