

---

## Cargols i femelles

P68660\_ca

---

Ens han caigut a terra  $n$  cargols i  $n$  femelles ( $n \geq 0$ ). Cada cargol té un ample diferent, cada femella té un ample diferent, i cada cargol encaixa únicament amb una femella.



Volem aparellar els cargols amb les femelles però, maularadament, l'única operació bàsica que podem fer és comparar un cargol amb una femella: el resultat pot ser que el cargol encaixa amb la femella, que el cargol és massa gran per la femella, o que el cargol és massa petit per la femella.

Primer, penseu com resoldre aquest problema en temps  $O(n \log n)$  en mitjana. Per a fer-ho, inspireu-vos en l'algorisme de *quick sort* i utilitzeu aquestes ajudes:

- Donat un cargol i algunes femelles, hi haurà una femella que encaixarà i que permetrà decidir si la resta de femelles són més grans o més petites.
- Igualment, donada una femella i alguns cargols, hi haurà un cargol que encaixarà i que permetrà decidir si la resta de cargols són més grans o més petits.

Un cop hagueu pensat com resoldre el problema, descarregueu el programa `code.cc` (icona `.cpp` al principi de l'enunciat) per implementar l'acció

```
void ordenar(const Cargols& cargols , const Femelles& femelles ,  
             Cargols& cargols_ordenats , Femelles& femelles_ordenades );
```

que, donat un vector d' $n$  cargols i un vector d' $n$  femelles (cada cargol amb un ample diferent, cada femella amb un ample diferent, i cada cargol encaixant únicament amb una femella), retorni un vector amb els cargols ordenats i un vector amb les femelles ordenades (en ordre creixent ambdós). Fixeu-vos que `cargols_ordenats` i `femelles_ordenades` són paràmetres de sortida (no d'entrada/sortida) i que ja teniu el cas base escrit.

La vostra acció `ordenar()` ha d'utilitzar la funció

```
int compara(Cargol cargol , Femella femella );
```

que compara un cargol amb una femella: si el cargol és més estret que la femella, retorna  $-1$ ; si el cargol encaixa amb la femella, retorna  $0$ ; i si el cargol és més ample que la femella, retorna  $1$ .

El tipus `Cargol` i el tipus `Femella` tenen un sol atribut que és el seu *ample*, però **heu de considerar que aquest atribut és privat**: no el podeu consultar ni modificar en el *vostre* codi (per senzillesa, el codi que us donem sí que l'utilitza).

El programa principal (ja escrit), s'encarrega de llegir els vectors de cargols i femelles, cridar a `ordenar()` i escriure els vectors ordenats.

## Entrada

L'entrada té diferents casos. Cada cas comença amb el nombre  $n$  de cargols i femelles i després venen els  $n$  amplex dels cargols i els  $n$  amplex de les femelles. Cada cargol té un ample diferent, cada femella té un ample diferent, i cada cargol encaixa únicament amb una femella. Els amplex dels cargols i de les femelles venen permutats a l'atzar.

## Sortida

La sortida és el vector de cargols i el vector de femelles ordenats (per tant, dos cops la mateixa informació) per a cada cas.

### Important:

- El vostre programa ha de començar amb un comentari que expliqui l'estratègia del funcionament del vostre algorisme.
- No podeu comparar cargols entre ells ni femelles entre elles. Només podeu comparar cargols amb femelles.
- Les funcions donades no es poden modificar. Podeu afegir noves funcions si ho considereu adient.
- No podeu utilitzar l'atribut *ample* dels cargols ni de les femelles.
- L'algorisme trivial  $O(n^2)$  no és una solució acceptable.

### Exemple d'entrada 1

```
8
 20 25 10 15 30 40 35 80
 80 20 10 25 15 40 35 30
```

### Exemple d'entrada 2

```
5
 3 18 2 5 7
 5 7 2 3 18
4
 1 5 2 6
 1 2 5 6
```

### Exemple de sortida 1

```
10 15 20 25 30 35 40 80
10 15 20 25 30 35 40 80
```

### Exemple de sortida 2

```
2 3 5 7 18
2 3 5 7 18
1 2 5 6
1 2 5 6
```

## Informació del problema

Autor : Jordi Petit

Generació : 2024-05-02 21:59:12

© [Jutge.org](https://jutge.org), 2006–2024.

<https://jutge.org>