
Mètode de Queue que duplica només els elements que pertanyen a un set S59319_ca

Donada la classe `Queue<T>` implementada amb una llista simplement encadenada, fes el mètode **RECURSIU**:

```
void duplicate(const set<T>& s);
```

Aquest mètode duplica cada element de la cua que pertanyi a *s*, és a dir, ha d'aparèixer dues vegades consecutives (original i la còpia, adjacents). Els elements que no pertanyen a *s* no es toquen. Si un element apareix diverses vegades a la cua i pertany a *s*, totes les seves ocurrències es dupliquen.

Exemple

cua entrada	3 1 4 1 5 9 2 6
set	{1, 5, 6}
cua sortida	3 1 1 4 1 1 5 5 9 2 6 6

Què has de lliurar

D'entre els fitxers que s'adjunten en aquest exercici, trobaràs `queue.old.hpp`, a on hi ha una implementació de la classe genèrica `Queue`. En primer lloc, hauràs de fer:

```
cp queue.old.hpp queue.hpp
```

A continuació si obres el fitxer `queue.hpp` al final del mateix trobaràs el mètode que has d'implementar:

```
void duplicate(const set<T>& s);
```

IMPORTANT: No toquis la resta de la implementació de la classe, excepte si per algun motiu, consideres que necessites afegir algun mètode auxiliar o atribut a la part privada.

D'entre els fitxers que s'adjunten a l'exercici també hi ha `program.cpp` (programa principal) i `Makefile` per a compilar i generar l'executable. El programa principal que t'oferim ja s'encarrega de llegir les cues i els sets i fer les crides al mètode indicat. **Només cal que implementis el mètode `duplicate`.**

Per a pujar la teva solució, has de crear el fitxer `solution.tar` així:

```
tar cf solution.tar queue.hpp
```

Observacions

- No es pot usar cap contenidor auxiliar de la STL (vector, stack, queue, ...) llevat de set.
- Has de trobar una solució **RECURSIVA** i eficient del problema. En particular, no hi hauria d'haver cap bucle en cap dels mètodes que implementis.
- Hauries d'aconseguir implementar el mètode demanat a base de crear nous nodes. De fet, una solució a base d'usar `push` i `pop` potser et permetrà passar els jocs de proves, però atès que la solució ha de ser eficient en temps i espai, aquest tipus de solució serà fortament penalitzat.
- Recorda que tots els mètodes que implementis han de tenir les corresponents **Precondició** (Pre) i **Postcondició** (Post).
- En les crides recursives inclou la **hipòtesi d'inducció** (HI) i la **funció de fita** (FF).

Entrada

L'entrada del programa és una seqüència de cues i sets.

Cada línia descriu un cas de prova amb el format:

$$[e_1 \ e_2 \ \dots \ e_N] \ \{s_1 \ s_2 \ \dots \ s_M\}$$

on $e_1 \dots e_N$ són els elements de la cua des del front al back, i $s_1 \dots s_M$ són els elements del set (tots diferents).

Sortida

Per cada cas de prova s'escriurà la cua resultant d'aplicar el mètode `duplicate`. Per escriure les cues, s'ha utilitzat l'operador `<<` que es troba definit en el fitxer `queue.hpp`.

Exemple d'entrada 1

```
[3 1 4 1 5 9 2 6] {1 5 6}
[1 2 3 4 5] {1 2 3 4 5}
[1 2 3 4 5] {6 7 8}
[7] {7}
[7] {3}
[2 2 3 2 4] {2 4}
[10 20 30] {1 2 10 30 100 200}
[] {10 9 8 7 6 5 4 3 2 1 0}
```

Exemple de sortida 1

```
12 3 1 1 4 1 1 5 5 9 2 6 6
10 1 1 2 2 3 3 4 4 5 5
5 1 2 3 4 5
2 7 7
1 7
9 2 2 2 2 3 2 2 4 4
5 10 10 20 30 30
0
```

Exemple d'entrada 2

```
[4 1 9 8 8 5 4 18 3 19 14 2 1 3 7 8 17 20]
[12 20 9 2 15 18 4 13 3 18 10 20 12 19 7]
[12 12 7 9 3 20 6 18 8 6 15 13 9 18 8] {2}
[11 7 16 13 15 5 9 5 8 18 18 9 19 14 19]
[18 1 4 18 9 11 4 10 14 6 15 1 9 17 6 17]
[16 1 4 12 10 8 2 8 19 3 3 16 3 18 5 5 16]
[15 4 8 8 3 11 1 19 18 8 19 8 1 3 2 8 3 2]
[7 4 4 14 12 14 14 15 2 4 2 13 11 4 8 7 7]
[14 16 16 7 13 2 6 13 1 13 9 15 10 14 18]
[2 19 16 17 17 6 2 17 3 6 3] {4 8 13 20}
```

Exemple de sortida 2

```
42 14 7 189 14 8 855 15 4 4 68 148 3214 54 72 92 11 357 178 8 17
32 2 82 102 32 8 9 {9 221 531 518 4 13 13 3 18 10 20 12 12 19
21 512 12 739 13 20} 20 6 18 18 8 8 6 15 13 13 9 18 18 8
33 3218 5 16 13 3214 55 65 14 208 1831859 179204 19 13 13 1
42 6018 {1 5 6 10 921 174 10 10 14 6 6 15 1 1 9 17 17 6 6 1
28 6691 4612 80100 18 832 18 881203 3 16 3 18 18 5 5 16 1
37 8514 8 8 868 73 181 181961981891208 8 19 19 8 8 1 3 2
38 754 54 14 6 94 12 24 314 8591512 15 464 127} 2 13 11 11 4 4
22 24516 176 97 10 12 282 10 6 13 1 1 13 9 9 15 10 10 14 18
11 2 19 16 17 17 6 2 17 3 6 3
```

Informació del problema

Autoria: Bernardino Casas

Generació: 2026-06-11T20:25:44.716Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>