
Afegir elements a una Queue només si continua siguent creixent. X18966_ca

En aquest exercici modificarem el mètode `push` de la classe `Queue` per tal d'assegurar que els seus elements estan sempre ordenats de forma estrictament creixent des del principi fins el final de la cua. El canvi és molt senzill: en una crida a `push`, si afegir l'element provocarà que la `Queue` deixi de ser estrictament creixent, llavors aquest element s'ignora i no s'afegeix.

Per exemple, suposeu que la `Queue` `q` conté els elements `[1,3,6]` (on els elements els representem en ordre des del `front` fins el final, i en particular `6` és l'últim element de la cua), i que es fa una crida `q.push(4)`. Llavors, `q` no haurà canviat i continuarà contenint `[1,3,6]`. Una crida `q.push(6)` tampoc provocarà cap canvi. Però una crida `q.push(7)` provocarà que `q` passi a contenir `[1,3,6,7]`.

D'entre els fitxers que s'adjunten en aquest exercici, trobareu `queue.hh`, a on hi ha una implementació de la classe genèrica `Queue`. Haureu de buscar dins `queue.hh` la implementació del mètode `push` i adaptar-lo convenientment.

Podeu suposar que el tipus genèric `T` de la classe té predefinida la operació de comparació `<`. Fins i tot, si voleu, poseu suposar que també teniu `>`, `<=`, `>=`, tot i que realment no cal. De fet, es testejarà la vostra implementació amb el tipus `T=int`. Ara bé, una solució que no sigui genèrica es considerarà incorrecta i serà invalidada a posteriori, encara que superi els jocs de proves.

D'entre els fitxers que s'adjunten a l'exercici també hi ha `main.cc` (programa principal), i el podeu compilar directament, doncs fa `include` de `queue.hh`. Només cal que pugueu `queue.hh` al jutge.

Entrada

L'entrada del programa comença amb una declaració d'unes quantes cues d'enters (`q0`, `q1`, ...), i després té una seqüència de comandes sobre les cues declarades. Com que ja us oferim el `main.cc`, no cal que us preocupeu d'implementar la lectura d'aquestes entrades. Només cal que implementeu la extensió de la classe cua abans esmentada.

Se suposa que la seqüència d'entrada serà correcta (sense `pop` ni `front` sobre cua buida).

El programa principal que us oferim ja s'encarrega de llegir aquestes entrades i fer les crides als corresponents mètodes de la classe cua. Només cal que feu els canvis abans esmentats.

Sortida

Per a cada comanda d'escriptura sobre la sortida s'escriurà el resultat corresponent. El `main.cc` que us oferim ja fa això. Només cal que implementeu la extensió de la classe cua abans esmentada.

Exemple d'entrada 1

```
Queue<int> q0 , q1 ;
q0 .push( 1 );
q0 .push( 3 );
q0 .push( 6 );
q1 .push( 6 );
q1 .push( 3 );
```

```
q1 .push( 1 );
cout<< q0 <<endl;
cout<< q1 <<endl;
cout<< q0 .size()<<endl;
cout<< q1 .size()<<endl;
cout<< q0 .front()<<endl;
cout<< q1 .front()<<endl;
```

```

q0 .push( 6 );
q1 .push( 6 );
cout<< q0 <<endl;
cout<< q1 <<endl;
cout<< q0 .size()<<endl;
cout<< q1 .size()<<endl;
cout<< q0 .front()<<endl;
cout<< q1 .front()<<endl;
q0 .push( 7 );
q1 .push( 7 );
cout<< q0 <<endl;
cout<< q1 <<endl;
cout<< q0 .size()<<endl;
cout<< q1 .size()<<endl;
cout<< q0 .front()<<endl;
cout<< q1 .front()<<endl;
q0 .push( 7 );
q1 .push( 7 );
cout<< q0 <<endl;
cout<< q1 <<endl;
cout<< q0 .size()<<endl;
cout<< q1 .size()<<endl;
cout<< q0 .front()<<endl;
cout<< q1 .front()<<endl;
q0 .pop();
q1 .pop();
cout<< q0 <<endl;
cout<< q1 <<endl;
cout<< q0 .size()<<endl;
cout<< q1 .size()<<endl;
cout<< q0 .front()<<endl;
cout<< q1 .front()<<endl;
q0 .pop();
q1 .pop();
cout<< q0 <<endl;
cout<< q1 <<endl;
cout<< q0 .size()<<endl;
cout<< q1 .size()<<endl;
q0 .push( -1 );
q1 .push( -1 );
cout<< q0 <<endl;
cout<< q1 <<endl;
cout<< q0 .size()<<endl;
cout<< q1 .size()<<endl;
cout<< q0 .front()<<endl;
cout<< q1 .front()<<endl;
q0 .push( 6 );
q1 .push( 6 );
cout<< q0 <<endl;
cout<< q1 <<endl;
cout<< q0 .size()<<endl;
cout<< q1 .size()<<endl;
cout<< q0 .front()<<endl;
cout<< q1 .front()<<endl;

```

Exemple d'entrada 2

```

Queue<int> q0 , q1 , q2 ;
cout<< q1 .size()<<endl;
cout<< q0 .size()<<endl;
q1 .push( -20 );

```

Exemple de sortida 1

```

1 3 6
6
3
1
1
6
1 3 6
6
3
1
1
6
1 3 6 7
6 7
4
2
1
6
1 3 6 7
6 7
4
2
1
6
3 6 7
7
3
1
3
7
6 7
2
0
6 7
-1
2
1
6
-1
6 7
-1 6
2
2
6
-1

```

```

q1 .pop();
q1 .push( 15 );
q2 .push( 41 );
q0 .push( 39 );
q2 .pop();

```

```

q0 .pop();
q2 .push( 15 );
q1 .push( 13 );
q2 .push( 17 );
cout<< q0 .size()<<endl;
q0 .push( -24 );
q1 .pop();
q0 .push( -19 );
cout<< q2 .size()<<endl;
q2 .push( 19 );
cout<< q2 .front()<<endl;
cout<< q1 .size()<<endl;
cout<< q2 .size()<<endl;
cout<< q2 .front()<<endl;
cout<< q0 .front()<<endl;
q2 .push( 17 );
cout<< q2 <<endl;
q0 .push( -20 );
cout<< q0 .front()<<endl;
q0 .pop();
cout<< q2 .front()<<endl;
q2 .pop();
q0 .push( -15 );
q0 .push( -16 );
q2 .push( 23 );
q2 .push( 21 );
q0 .push( -12 );
q0 .push( -8 );
q1 .push( -12 );
q2 .push( 26 );
q2 .push( 28 );
q0 .push( -9 );
cout<< q0 .size()<<endl;
q1 .push( -16 );
cout<< q2 .size()<<endl;
q1 .push( -11 );
q1 .pop();
q1 .push( -14 );
q1 .push( -6 );
q0 .push( -11 );
cout<< q1 .size()<<endl;
q1 .pop();
q0 .pop();
q2 .push( 33 );
q2 .push( 35 );
q0 .pop();
cout<< q2 .size()<<endl;
q2 .push( 39 );
q1 .pop();
q1 .push( -46 );
cout<< q2 .size()<<endl;
q1 .push( -46 );
q1 .push( -47 );
q1 .push( -43 );
cout<< q1 .size()<<endl;
q1 .push( -39 );
q1 .push( -39 );
cout<< q2 .front()<<endl;
cout<< q0 <<endl;
q2 .push( 36 );
cout<< q1 .front()<<endl;

```

```

q2 .pop();
cout<< q1 .size()<<endl;
q2 .push( 42 );
q2 .pop();
q2 .push( 45 );
q1 .push( -40 );
q0 .push( -6 );
q0 .pop();
q1 .pop();
q1 .push( -38 );
q2 .push( 48 );
q1 .pop();
q0 .pop();
q0 .pop();
q0 .push( 47 );
q0 .push( 49 );
q2 .push( 45 );
cout<< q1 .size()<<endl;
cout<< q2 .size()<<endl;
q2 .push( 48 );
cout<< q2 .size()<<endl;
q2 .push( 51 );
cout<< q0 .front()<<endl;
cout<< q2 .size()<<endl;
cout<< q2 .front()<<endl;
cout<< q2 .size()<<endl;
q1 .push( -42 );
cout<< q1 .size()<<endl;
q1 .push( -40 );
q1 .pop();
cout<< q1 .front()<<endl;
q0 .pop();
cout<< q1 <<endl;
cout<< q0 .front()<<endl;
q2 .pop();
q0 .pop();
q2 .push( 46 );
q2 .push( 54 );
cout<< q2 .size()<<endl;
q0 .push( 46 );
q2 .push( 59 );
q0 .push( 41 );
cout<< q2 .size()<<endl;
cout<< q0 .size()<<endl;
q2 .push( 56 );
cout<< q0 .size()<<endl;
q1 .push( -36 );
cout<< q1 .size()<<endl;
q2 .push( 58 );
q1 .push( -40 );
cout<< q0 <<endl;
cout<< q1 <<endl;
cout<< q0 .size()<<endl;
q0 .push( 51 );
cout<< q1 .size()<<endl;
q0 .push( 52 );
cout<< q2 <<endl;
cout<< q2 .size()<<endl;
q0 .push( 49 );
q1 .push( -37 );
cout<< q1 .front()<<endl;

```

```

q2 .pop();
cout<< q1 .front()<<endl;
cout<< q2 .size()<<endl;
q0 .pop();
q2 .pop();
cout<< q1 <<endl;
cout<< q1 <<endl;
cout<< q2 .size()<<endl;
cout<< q1 .size()<<endl;
q2 .push( 56 );
cout<< q2 .size()<<endl;
q1 .push( -37 );
cout<< q1 .size()<<endl;
cout<< q2 <<endl;
q2 .push( 64 );
q0 .push( 57 );
cout<< q0 .front()<<endl;
cout<< q1 .front()<<endl;
cout<< q1 .size()<<endl;
q1 .push( -40 );
cout<< q0 .front()<<endl;
cout<< q0 <<endl;
q1 .pop();
q2 .push( 65 );
q1 .pop();
cout<< q1 .size()<<endl;
q2 .pop();
q0 .push( 59 );
q1 .push( -20 );
cout<< q1 <<endl;
q0 .push( 59 );
cout<< q1 .size()<<endl;
cout<< q0 .size()<<endl;
cout<< q0 .front()<<endl;
q1 .pop();
q0 .pop();
cout<< q1 .size()<<endl;
cout<< q0 <<endl;
q1 .push( -6 );
cout<< q0 <<endl;
cout<< q0 .size()<<endl;
q2 .push( 64 );
q1 .push( -2 );
cout<< q2 .front()<<endl;
q1 .push( -6 );
cout<< q2 .size()<<endl;
q1 .push( 0 );
cout<< q0 .size()<<endl;
q0 .push( 62 );
cout<< q0 .front()<<endl;
cout<< q2 <<endl;
q2 .push( 60 );
cout<< q2 .size()<<endl;
q0 .push( 58 );
cout<< q2 .front()<<endl;
cout<< q0 .front()<<endl;
cout<< q2 .front()<<endl;
q2 .pop();
q1 .push( 0 );
q2 .pop();
cout<< q2 .front()<<endl;

```

```

q0 .pop();
q2 .pop();
cout<< q0 .front()<<endl;
q1 .push( 1 );
q0 .push( 60 );
cout<< q2 .size()<<endl;
cout<< q0 .size()<<endl;
q1 .pop();
cout<< q2 .size()<<endl;
q0 .pop();
cout<< q0 <<endl;
cout<< q1 <<endl;
cout<< q2 <<endl;

```

Exemple de sortida 2

0	26 28 33 35 39 42 45 48 51 54 59
0	11
0	-38
2	-38
15	10
0	-38 -36
3	-38 -36
15	9
-24	2
15 17 19	9
-24	2
15	33 35 39 42 45 48 51 54 59
4	51
5	-38
2	2
7	51
8	51 52 57
2	0
17	-20
-12 -8	1
-46	4
3	51
2	0
9	52 57 59
9	52 57 59
47	3
10	35
23	10
10	3
2	52
-38	35 39 42 45 48 51 54 59 64 65
-38	10
49	35
10	52
11	35
1	42
1	57
2	7
46	3
-38 -36	7
1	59 62
2	-2 0 1
	45 48 51 54 59 64 65

Observació

Avaluació sobre 10 punts: (Afegiu comentaris si el vostre codi no és prou clar)

- Solució lenta: 5 punts.
- solució ràpida: 10 punts.

Entenem com a solució ràpida una que és correcta i capaç de superar els jocs de proves públics i privats. Entenem com a solució lenta una que no és ràpida, però és correcta i capaç de superar els jocs de proves públics.

Una solució que no sigui genèrica (per a qualsevol tipus T amb $j, \zeta, j=, \zeta=$) serà invalidada i rebrà nota 0, encara que superi els jocs de proves.

Una solució que crei memòria innecessàriament rebrà una penalització, i aquesta penalització serà fins i tot major si a sobre aquesta memòria innecessària no s'allibera, encara que superi els jocs de proves.

Informació del problema

Autor : PRO2

Generació : 2024-04-23 13:44:45

© *Jutge.org*, 2006–2024.

<https://jutge.org>