Afegir elements a un Stack només si continua siguent creixent.X24836_ca

En aquest exercici modificarem el mètode `push` de la classe `Stack` per tal d'assegurar que els seus elements estan sempre ordenats de forma estrictament creixent des del fons fins el top de la pila. El canvi és molt senzill: en una crida a `push`, si afegir l'element provocarà que l'`Stack` deixi de ser estrictament creixent, llavors aquest element s'ignora i no s'afegeix.
Per exemple, suposeu que l'`Stack s` conté els elements $[1,3,6]$ (on els elements els representem en ordre des del fons fins el top, i en particular 6 és l'element del top de la pila), i que es fa una crida `s.push(4)`. Llavors, `s` no haurà canviat i continuarà contenint $[1,3,6]$. Una crida `s.push(6)` tampoc provocarà cap canvi. Però una crida `s.push(7)` provocarà que `s` passi a contenir $[1,3,6,7]$.
D'entre els fitxers que s'adjunten en aquest exercici, trobareu `stack.hh`, a on hi ha una implementació de la classe genèrica `Stack`. Haureu de buscar dins `stack.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 `stack.hh`. Només cal que pugeu `stack.hh` al jutge.

Entrada

L'entrada del programa comença amb una declaració d'unes quantes piles d'enters (`s0, s1, ...`), i després té una seqüència de comandes sobre les piles 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 pila abans esmentada.
Se suposa que la seqüència d'entrada serà correcta (sense pop ni top sobre pila buida).
El programa principal que us oferim ja s'encarrega de llegir aquestes entrades i fer les crides als corresponents mètodes de la classe pila. 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 pila abans esmentada.

Exemple d'entrada 1

```
Stack<int> s0 , s1 ;
s0 .push( 1 );
s0 .push( 3 );
s0 .push( 6 );
s1 .push( 6 );
s1 .push( 3 );
s1 .push( 1 );
cout<< s0 <<endl;
cout<< s1 <<endl;
cout<< s0 .size()<<endl;
cout<< s1 .size()<<endl;
cout<< s0 .top()<<endl;
cout<< s1 .top()<<endl;
s0 .push( 6 );
s1 .push( 6 );
cout<< s0 <<endl;
```

```
cout<< s1 <<endl;
cout<< s0 .size()<<endl;
cout<< s1 .size()<<endl;
cout<< s0 .top()<<endl;
cout<< s1 .top()<<endl;
s0 .push( 7 );
s1 .push( 7 );
cout<< s0 <<endl;
cout<< s1 <<endl;
cout<< s0 .size()<<endl;
cout<< s1 .size()<<endl;
cout<< s0 .top()<<endl;
cout<< s1 .top()<<endl;
s0 .push( 7 );
s1 .push( 7 );
cout<< s0 <<endl;
cout<< s1 <<endl;
cout<< s0 .size()<<endl;
cout<< s1 .size()<<endl;
cout<< s0 .top()<<endl;
cout<< s1 .top()<<endl;
s0 .pop();
s1 .pop();
cout<< s0 <<endl;
cout<< s1 <<endl;
cout<< s0 .size()<<endl;
cout<< s1 .size()<<endl;
cout<< s0 .top()<<endl;
cout<< s1 .top()<<endl;
s0 .pop();
s1 .pop();
cout<< s0 <<endl;
cout<< s1 <<endl;
cout<< s0 .size()<<endl;
cout<< s1 .size()<<endl;
s0 .push( -1 );
s1 .push( -1 );
cout<< s0 <<endl;
cout<< s1 <<endl;
cout<< s0 .size()<<endl;
cout<< s1 .size()<<endl;
cout<< s0 .top()<<endl;
cout<< s1 .top()<<endl;
s0 .push( 6 );
s1 .push( 6 );
cout<< s0 <<endl;
cout<< s1 <<endl;
cout<< s0 .size()<<endl;
cout<< s1 .size()<<endl;
cout<< s0 .top()<<endl;
cout<< s1 .top()<<endl;
```

Exemple de sortida 1

```
1 3 6
6
3
1
6
6
1 3 6
6
3
1
6
6
1 3 6 7
6 7
4
2
7
7
1 3 6 7
6 7
4
2
7
7
1 3 6
6
3
1
6
6
1 3

2
0
1 3
-1
2
1
3
-1
1 3 6
-1 6
3
2
6
6
```

Exemple d'entrada 2

```
Stack<int> s0 , s1 , s2 ;
cout<< s1 .size()<<endl;
cout<< s2 .size()<<endl;
s1 .push( -20 );
s1 .pop();
s1 .push( 15 );
s2 .push( 41 );
s0 .push( 39 );
s2 .pop();
s0 .pop();
s2 .push( 15 );
s1 .push( 13 );
s2 .push( 17 );
s0 .push( -24 );
s1 .pop();
```

```
s0 .push( -19 );
cout<< s2 .top()<<endl;
s2 .push( 19 );
cout<< s2 .size()<<endl;
cout<< s2 .top()<<endl;
cout<< s2 .size()<<endl;
cout<< s1 .size()<<endl;
cout<< s0 .size()<<endl;
s2 .push( 17 );
cout<< s2 <<endl;
s0 .push( -20 );
cout<< s0 .size()<<endl;
s0 .pop();
cout<< s2 .size()<<endl;
s2 .pop();
s0 .push( -20 );
s0 .push( -21 );
s2 .push( 21 );
s2 .push( 19 );
s0 .push( -17 );
s0 .push( -13 );
s1 .push( -12 );
s2 .push( 24 );
s2 .push( 26 );
s0 .push( -14 );
cout<< s0 .top()<<endl;
s1 .push( -16 );
cout<< s2 .top()<<endl;
s1 .push( -11 );
s1 .pop();
s1 .push( -15 );
s1 .push( -7 );
s0 .push( -16 );
cout<< s1 .top()<<endl;
s1 .pop();
s0 .pop();
s2 .push( 31 );
s2 .push( 33 );
s0 .pop();
cout<< s2 .top()<<endl;
s2 .push( 37 );
s1 .pop();
s1 .push( -46 );
cout<< s2 .top()<<endl;
s1 .push( -46 );
s1 .push( -47 );
s1 .push( -43 );
cout<< s1 .top()<<endl;
s1 .push( -39 );
s1 .push( -39 );
cout<< s2 .size()<<endl;
cout<< s0 <<endl;
s2 .push( 34 );
cout<< s1 .size()<<endl;
s2 .pop();
cout<< s1 .top()<<endl;
s2 .push( 36 );
s2 .pop();
s2 .push( 36 );
s1 .push( -40 );
s0 .push( -18 );
s0 .pop();
s1 .pop();
s1 .push( -42 );
s2 .push( 39 );
s1 .pop();
s0 .pop();
s0 .pop();
s0 .push( 47 );
s0 .push( 49 );
s2 .push( 36 );
cout<< s1 .top()<<endl;
cout<< s2 .top()<<endl;
s2 .push( 39 );
cout<< s2 .top()<<endl;
s2 .push( 42 );
cout<< s0 .size()<<endl;
cout<< s2 .top()<<endl;
cout<< s2 .size()<<endl;
cout<< s2 .top()<<endl;
s1 .push( -47 );
cout<< s1 .top()<<endl;
s1 .push( -45 );
s1 .pop();
cout<< s1 .size()<<endl;
s0 .pop();
cout<< s1 <<endl;
cout<< s0 .size()<<endl;
s2 .pop();
s0 .pop();
s2 .push( 34 );
s2 .push( 42 );
cout<< s2 .top()<<endl;
s0 .push( 46 );
s2 .push( 47 );
s0 .push( 41 );
cout<< s2 .top()<<endl;
cout<< s0 .top()<<endl;
s2 .push( 44 );
cout<< s0 .top()<<endl;
s1 .push( -44 );
cout<< s1 .top()<<endl;
s2 .push( 46 );
s1 .push( -48 );
cout<< s0 <<endl;
cout<< s1 <<endl;
cout<< s0 .top()<<endl;
s0 .push( 51 );
cout<< s1 .top()<<endl;
s0 .push( 52 );
cout<< s2 <<endl;
cout<< s2 .top()<<endl;
s0 .push( 49 );
s1 .push( -45 );
cout<< s1 .size()<<endl;
s2 .pop();
cout<< s1 .size()<<endl;
cout<< s2 .top()<<endl;
s0 .pop();
s2 .pop();
cout<< s1 <<endl;
cout<< s1 <<endl;
```

```
cout<< s2 .top()<<endl;
cout<< s1 .top()<<endl;
s2 .push( 36 );
cout<< s2 .top()<<endl;
s1 .push( -45 );
cout<< s1 .top()<<endl;
cout<< s2 <<endl;
s2 .push( 44 );
s0 .push( 56 );
cout<< s0 .size()<<endl;
cout<< s1 .size()<<endl;
cout<< s1 .top()<<endl;
s1 .push( -48 );
cout<< s0 .size()<<endl;
cout<< s0 <<endl;
s1 .pop();
s2 .push( 45 );
s1 .pop();
s2 .pop();
s0 .push( 58 );
s1 .push( -20 );
cout<< s1 <<endl;
s0 .push( 58 );
cout<< s1 .top()<<endl;
cout<< s0 .top()<<endl;
cout<< s0 .size()<<endl;
s1 .pop();
s0 .pop();
cout<< s0 <<endl;
s1 .push( -6 );
cout<< s0 <<endl;
cout<< s0 .top()<<endl;
s2 .push( 43 );
s1 .push( -2 );
cout<< s2 .size()<<endl;
s1 .push( -6 );
cout<< s2 .top()<<endl;
s1 .push( 0 );
cout<< s0 .top()<<endl;
s0 .push( 59 );
cout<< s0 .size()<<endl;
cout<< s2 <<endl;
s2 .push( 39 );
cout<< s2 .top()<<endl;
s0 .push( 55 );
cout<< s2 .size()<<endl;
cout<< s0 .size()<<endl;
cout<< s2 .size()<<endl;
s2 .pop();
s1 .push( 0 );
s2 .pop();
cout<< s2 .size()<<endl;
s0 .pop();
s2 .pop();
cout<< s0 .size()<<endl;
s1 .push( 1 );
s0 .push( 54 );
cout<< s2 .top()<<endl;
cout<< s0 .top()<<endl;
s1 .pop();
cout<< s2 .top()<<endl;
s0 .pop();
s1 .push( -4 );
s0 .push( 47 );
cout<< s2 .size()<<endl;
cout<< s0 <<endl;
cout<< s1 <<endl;
cout<< s2 <<endl;
```

Exemple de sortida 2

```
0
0
17
3
19
3
0
2
15 17 19
2
3
-13
26
-7
33
37
-43
8
-24 -20
3
-39
-43
39
39
2
42
10
42
-43
1
-46
1
42
47
46
46
-44
46
-46 -44
46
-44
15 17 21 24 26 31 33 36 39 42 47
47
2
2
42
-46 -44
-46 -44
39
-44
39
-44
15 17 21 24 26 31 33 36 39
3
2
-44
3
46 51 56
-20
-20
58
4
46 51 56
46 51 56
56
10
44
56
4
15 17 21 24 26 31 33 36 39 44
44
10
4
10
8
3
33
56
33
7
46 51
-6 -2 0
15 17 21 24 26 31 33
```

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 <, >, <=, >=) 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

Autoria: PRO2

Generació: 2026-01-27T18:51:50.069Z