

Cvičná úloha 07

Termín odevzdání: 10.05.2009 23:59:59 531582.103 sec

Hodnocení: 0.0000

Vybraná / zadaná úloha: Množina

Úkolem je navrhnout a implementovat třídu `Mnozina`, která bude realizovat množinu celých čísel bez duplicit pomocí vzestupně uspořádaného a rozšiřitelného pole.

Rozhraní třídy musí obsahovat:

- Implicitní konstruktor, který vytvoří prázdnou množinu.
- Kopírující konstruktor, který provede inicializaci deklarovaného objektu hlubokou kopií.
- Destruktor, který uvolní alokovanou paměť.
- Metodu `bool obsahuje (int x)`, která zjistí, zda hodnota `x` je v množině obsažena (vrací `true`) nebo není (vrací `false`).
- Metodu `vloz (int x)`, která do množiny vloží číslo `x`, pokud v ní toto číslo ještě není obsaženo. Metodu deklaruje a definuje tak, aby například vložení čísel 3 a 4 do množiny `m` bylo možné zapsat příkazem:

```
m.vloz(3).vloz(4);
```

- Metodu `odeber (int x)`, která z množiny odebere číslo `x`. Pokud v množině odebírané číslo není, množina se nezmění. Metodu deklaruje a definuje tak, aby například odebrání čísel 3 a 4 z množiny `m` bylo možné zapsat příkazem:

```
m.odeber(3).odeber(4);
```

- Přetížení operátoru přiřazení (`=`), který provede hlubokou kopii.
- Přetížení operátoru `*` tak, aby výsledkem operace:

```
m1 * m2
```

byla množina, která je průnikem množin `m1` a `m2`. Při realizaci využijte toho, že množiny jsou realizované uspořádaným polem, takže složitost této operace bude $O(n+m)$, kde `n` a `m` jsou mohutnosti množin `m1` a `m2`. Operace vytváří novou instanci, parametry (zde množiny `m1` a `m2`) nejsou operací měněny.

- Přetížení operátor `<<` pro výstupní konverzi do datového proudu. Výstupem bude do hranatých závorek `[a]` uzavřená vzestupně seřazená posloupnost čísel obsažených v množině a oddělovaných čárkou. Přetížený operátor samotný nebude do výstupu přidávat odřádkování.

Odevzdávejte soubor, který obsahuje pouze implementovanou třídu `Mnozina`. Třída musí splňovat popsané rozhraní - jinak dojde k chybě při kompilaci. Do třídy si ale můžete doplnit další metody (veřejné nebo i privátní) a členské proměnné. Odevzdávaný soubor musí obsahovat jak deklaraci třídy (popis rozhraní) tak i definice metod, konstruktoru a destrukturu. Je jedno, zda jsou metody implementované inline nebo odděleně. Odevzdávaný soubor nesmí kromě implementace třídy `Mnozina` obsahovat nic jiného, zejména ne funkci `main`. Pro usnadnění odevzdávání můžete využít toho, že ProgTest definuje proměnnou preprocesoru `__PROGTEST__`.

Implementovaná třída nesmí používat datové struktury z STL (`vector`, `list`, ...). Jejich použití povede k chybě při kompilaci. Třída je testovaná v omezeném prostředí, kde je limitovaná dostupná paměť (dostačuje k uložení prvků v binární podobě) a je omezena dobou běhu (20s na testovacím počítači pro cca 50000 vkládaných/odebíraných prvků, referenční řešení pro výpočet potřebuje cca 1.5sec).

Ukázka použití třídy (v komentářích za výstupní konverzí je uveden požadovaný výstup):

```
Mnozina m1;
m1.vloz(19).vloz(10).vloz(2).vloz(19);
Mnozina m2;
m2 = m1;
m1.odeber(3).odeber(10);
Mnozina m3 = m2;
m3.vloz(15);
cout << m1 << endl; // [2,19]
cout << m2 << endl; // [2,10,19]
cout << m3 << endl; // [2,10,15,19]
```

```
Mnozina m4;
m4 = m1*m2;
cout << m4 << endl; //[2,19]
const Mnozina m0;
m2 = m1 = m0;
cout << m1 << endl; //[]
cout << m2 << endl; //[]
for (int i=-2; i<=5000; i++) {
    m1.vloz(i);
    m2.vloz(-i);
}
m3 = m1 * m2;
cout << m3 << endl; //[-2,-1,0,1,2]
cout << m0 << endl; //[]
```

Nápověda:

- Při vkládání neztvětšujte pole pole po jednom prvku. Zvolte vhodný inkrement (např. 1000 prvků najednou) nebo velikost pole zvětšujte geometrickou řadou (např. 2x).

Odevzdaná řešení: 5 / 10**Details****Hodnocení:** 0.0000