**Feladat:**

Valósítsa meg az egész számokat tartalmazó halmaz típust! Ábrázolja a halmaz

elemeit egy sorozatban! Implementálja a szokásos műveleteket (adott elem betevése,

kivétele, üres-e a halmaz), valamint a halmaz legnagyobb elemének kiválasztását

(ehhez célszerű a halmazt reprezentáló sorozat mellett külön eltárolni a legnagyobb

elemet, és ezt karbantartani), továbbá egy halmaz kiírását!

**Halmaz típus:**

A feladat lényege egy felhasználói típusnak, a halmaz típusnak a megvalósítása.A halmazban az elemek, amiket tárolunk, csak egyszer fordulhatnak elő.

Formálisan: *halmaz*={ v: vektor I {minden i,j [0..n]:v[i]!=v[j]: i!=j} }

**Típusérték-halmaz:**

A halmazban csak egész számokat tárolunk.

**Típus-műveletek:**

1.Elem betevése: Az adott egész szám hozzáadása a halmazhoz, amennyiben még nincs benne, majd a halmaz maximumának beállítása

2.Elem kivétele: Az adott egész szám kivétele a halmazból, emennyiben benne van, majd a halmaz maximumának beállítása

3.Üres-e a halmaz: Egy logikai értéket ad vissza aszerint, hogy a halmaz elemszáma

nulla-e.

4.Halmaz legnagyobb elemének kiválasztása: A halmaz legnagyobb elemét adja vissza, amennyiben nem üres a halmaz

**Reprezentáció:**

Egy n elemű halmazt, egy 0-tól n-1ig indexelt vektorral ábrázolunk.

**Implementáció:**

1.Elem betevése - add(a): A v vektorral ábrázolt halmazba beszúrjuk az a elemet, amennyiben még nincs benne, ezt az it iterátor segítségével nézzük meg, majd a set\_max() függvénnyel beállítjuk a halmaz legnagyobb elemét

|  |  |
| --- | --- |
| it:= v.keres(a) | |
| it < v.size() | |
| HIBA | v[v.size()+1]:=a;  set\_max(); |

2.Elem kivétele: remove(a): A v vektorral ábrázolt halmazból eltávolítjuk az a elemet, amennyiben benne van, amit az it iterátor segítségével nézünk meg ,majd a set\_max() függvénnyel beállítjuk a halmaz legnagyobb elemét

|  |  |
| --- | --- |
| it:= v.keres(a) | |
| it > v.size() | |
| HIBA | v[it]:=a;  set\_max(); |

3.Üres-e a halmaz: is\_empty() Egy L logikai értéket ad vissza aszerint, hogy a v vektorral ábrázolt halmaz elemszáma nulla-e.

|  |
| --- |
| L := v.size() == 0 |
| return L; |

4.Halmaz legnagyobb elemének kiválasztása: get\_max() : A set\_max() művelet segítségével beállítjuk a v vektorral ábrázolt halmaz legnagyobb elemét.

|  |
| --- |
| return max; |

set\_max():

|  |  |  |  |
| --- | --- | --- | --- |
| i=0..v.size()-2 |  | | |
| index=i; maximum=v[i] | | |
| j = i+1 .. v.size()-1 |  | |
| v[j] > maximum | |
| maximum = v[j] |  |
| index=j |  |
|  | v[index]= v[i]; v[i]=maximum | | |
| max=v[0] | | | |

**Osztály:**

A halmaz típust egy osztály segítségével valósítjuk meg. A konstruktorban meg lehet adni egy vektorral a halmaz elemeit, ha nincs megadva a konstruktorban, akkor üres halmazt hozunk létre. A maximum keresésnél, elem eltávolításánál meg kell vizsgálni, hogy nem üres-e a halmaz, ha igen akkor kivételt dobunk.

|  |
| --- |
| **Set** |
| - set: vector<int>  - max: int  - set\_max() : void |
| + Set()  + Set(vector<int>)  +is\_empty() : bool  + get\_max() : int  + add(int) : void  + remove(int) : void  +get\_size() : int  +print() : void |

Az osztály privát részében definiáljuk a vektort ami a halmaz elemeit tárolja. Ezen kívül privát adattag a max , és a set\_max függvény.

Az osztály publikus metódusai a kétféle konstruktor. Az egyik egy üres halmazt hoz létre, a másik pedig átmásolja a halmazba a megadott paramétert. További műveletek egy adott elem betevése és eltávolítása, a halmaz kiírása, és az üresség vizsgálat. Ezen kívül a get\_size() műveletet használjuk arra, hogy a tesztesetekben ellenőrizzük, hogy a betevés és az eltávolítás sikeres volt.

**Tesztelési terv:**

Megvalósított műveletek tesztelése

1. Különféle méretű halmazok létrehozása

2. Elemek beszúrása halmazokba

3. Elemek eltávolítása halmazokból

4. Maximum lekérdezése

5. Halmaz ürességének a vizsgálata

Tesztelés kód alapján

1. Halmaz ürességének vizsgálata elemek hozzáadása és eltávolítása után
2. Maximum tesztelése elemek eltávolítása és hozzáadásával
3. Elemek hozzáadása a halmazhoz
4. Elemek eltávolítása