

## 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!*

## Egész számokat tartalmazó halmaz típus

A feladat lényege egy felhasználói típusnak, az egész számokat tartalmazó halmaz típusnak a megvalósítása.

### Típusérték-halmaz:

Olyan egész számokat tartalmazó halmazzal szeretnénk dolgozni, amelyben egy elem csak egyszer fordulhat elő. Emellett eltároljuk az aktuális maximális értéket a halmazból.

### Típus-műveletek:

#### 1. Adott elem betevése:

A halmaz  $n+1$ -edik helyére ( $n$  a halmaz aktuális mérete) egy új elem betétele akkor, ha ez az elem még nem található meg a halmazban, és abban az esetben, ha az új elem nagyobb az eddigi maximumnál, a maximum átállítása.

#### 2. Adott elem kivétele:

Egy megadott elemet kiveszünk a halmazból, ha ez az elem szerepel a halmazban, illetve, ha ez az elem a maximum elem, akkor új maximumot is keresünk.

#### 3. Üres-e a halmaz:

Megvizsgáljuk, hogy a halmaz tartalmaz-e legalább egy elemet.  $a.size() > 0$ .

#### 4. Maximum elem:

Visszaadjuk a halmazból a legnagyobb elemet, amelyet külön tárolunk.

## Reprezentáció

Egy egész számokból álló halmazt kell ábrázolni, amit egy vektor (vec) segítségével 0-tól indexelve tárolunk el, emellett számontartjuk a halmaz legnagyobb elemét (max), amely természetesen egész szám.

## Implementáció

### 1. Adott elem betevése:

A vec vektorral ábrázolt halmaz  $n+1$ -edik helyére való új elem (e) beszúrása, abban az esetben, ha ez nem fordult még elő és egyben a legnagyobb elem(max) felülvizsgálata.

!vec.find(e)	
e > max	SKIP
max = num	SKIP
vec[n+1] = e	

### 2. Adott elem kivétele:

A vec vektorral ábrázolt halmazból egy e elem kivehető, ha e benne van vec-ben. Ha e elem egyenlő a legnagyobb értékkel, akkor egy új maximumot kell keresni.

vec.find(e)	
e == max	SKIP
tmpMax := e[0]	SKIP
i=1..n	
vec[i] > tmpMax && vec[i] < max	
tmpMax := vec[i]	SKIP
max := tmpMax	
vec[vec.find(e)].erase(e)	

### 3. Üres-e a halmaz:

A vec vektorral ábrázolt halmazról eldöntjük, hogy üres-e.

vec.size() == 0	
return true	return false

### 4. Maximum-elem:

A vec vektorral ábrázolt halmazból visszaadjuk a legnagyobb elemét, amit külön eltároltunk.

return max
------------

# Tesztelési terv

- 1) Elem hozzáadása a halmazhoz:
  - a) üres halmazhoz manuálisan hozzáadni az elemeket
  - b) a halmazt fájlból feltölteni
  - c) olyan elemet hozzáadni, ami már van a halmazban
- 2) Maximum elem vizsgálata:
  - a) üres halmazhoz, ha hozzáadunk egy elemet, az az egy elem lesz-e a maximum
  - b) 2 tagú halmazból, az utoljára berakott elem nagyobb, mint az első, az lesz-e a maximum
  - c) fájlból való beolvasás után, megfelelő-e a maximum
  - d) ha töröljük az aktuális maximumot, megfelelően beállítjuk-e az utána következőt
- 3) Üres-e a halmaz vizsgálata:
  - a) üres halmaz létrehozásakor, üres-e
  - b) üres halmazhoz hozzáadunk egy elemet, nem üres-e
  - c) eltávolítjuk a halmazból az összes elemet, üres-e
- 4) Elem törlése a halmazból:
  - a) üres halmazból való törlés
  - b) olyan számot törölni, ami van a halmazban
  - c) a maximum elemet törölni a halmazból, a maximum elem változik-e
  - d) olyan számot törölni, ami nincs a halmazban
- 5) Kivételek generálása és elkapása