**Molnár Dániel**

BMA0B9

[bma0b9@inf.elte.hu](mailto:bma0b9@inf.elte.hu) 3. csop

**1. beadandó/0.feladat**

2020. február 6.

# 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.

## Típusérték-halmaz1

Olyan számokat (ebben az esetben egész számokat: ℤ) tartalmazó *n*-elemű halmazt akarunk megvalósítani, amelyben egy egész szám csak egyszer szerepel. Az *n*ℕ a container méretét határozza meg.

Formálisan: *Set*(*n*) = { v ℤn  *i*[*1*..*n*]: *egyedi* ℤ (minden elem különböző)}

## Típus-műveletek2

1. *Betevés -pushBack()*

Az adott elem *e* halmazban való elhelyezése, ha még nem szerepel benne.

Formálisan: *A* : *Set*(*n*) × ℤ x ℤ

*a e max*

*Ef* : ( *a*=*a’*  *e=e’*  max=max’ )

*Uf* : ( ea, akkor Ef e∉a, akkor *a[n+1] = e* ha e > max, akkor max = e)

1. *Kivétel – erase()*

Az adott elem *e* halmazban való kivétele, ha szerepel benne.

Formálisan: *A* : *Set*(*n*) × ℤ x ℤ

*a e max*

*Ef* : ( *e*=*e’*  *a*=*a’*  ax=max’)

*Uf* : (ea, akkor a elemeinek száma csökken, illetve ha e == a legnagyobb számmal (max) akkor max-ot újra ki kell értékelni e∉a, akkor Ef)

1 A típusérték-halmazt kétféleképpen is le lehet írni: szövegesen és formálisan. Elég csak az egyik formát használni.

2 A típusműveletek leírására is kétféle definíciót használok: egy informálist és egy formálist. Elég csak az egyik formát használni.

1. *Üres-e – isEmpty()*

A halmaz üres-e – üres *true*, ellenben *false*

Formálisan: *A* : *Set*(*n*)

*a*

*Ef* : ( *a*=*a’* )

*Uf* = ( *Ef* )

1. *Legnagyobb elem lekérdezése -maxElement()*

A halmaz legnagyobb elemének kiválasztása

Formálisan: *A* : *Set(n)* × ℤ

a max

*Ef* : ( *a*=*a’*  *max*=*max’*)

*Uf* : ( Ha a halmaz nem üres akkor *Ef*)

## Reprezentáció

A halmaz elemei tetszőleges egész számok, de csak egyszer szerepelhetnek.

Pl. {1, 3, 5, 3};

## Implementáció3

1. *pushBack*

A *v* vectorral ábrázolt halmazban, ha már benne van a hozzáadni kívánt elem, akkor nem adjuk hozzá, ellenben hozzáadjuk és ha nagyobb mint a maximum element akkor értékül adjuk.

FÜGGVÉNY pushBack(int e)

HA v-ben bennevan e, akkor

return false

HA VISZONT v üres vagy e > mint max, akkor

max = e

ELÁGAZÁS VÉGE

v hez hozzáadjuk e-t

FÜGGVÉNY VÉGE

1. *erase*

A v vectorral ábrázolt halmazból – amennyiben benne van – kivesszük a megadott elemet.

FÜGGVÉNY erase(int e)

HA v üres-e, akkor

return false

c = iterátor, illetve érték páros (hol helyezkedik el az adott szám, mi az értéke)

ELÁGAZÁS VÉGE

HA iterátor valid, akkor

wasMax = ha a törölni kívánt elem megegyezik a legnagyobb számmal akkor true

e kitörlése v ből

HA v üres-e, akkor

max-ot resetelni kell és return true;

ELÁGAZÁS VÉGE

HA wasMax, akkor

újra ki kell értékelni max-ot

ELÁGAZÁS VÉGE

return true;

ELÁGAZÁS VÉGE

return false;

FÜGGVÉNY VÉGE

1. *isEmpty*

FÜGGVÉNY isEmpty()

HA v üres halmaz

return true

ELÁGAZÁS VÉGE

return false

FÜGGVÉNY VÉGE

1. *maxElement*

*A legnagyobb értékű elem visszaadása, ha nem üres halmaz*

FÜGGVÉNY maxElement()

HA max inicializálva van

return max értékét

ELÁGAZÁS VÉGE

Hiba kiírása – a halmaz üres

return {}

3 A műveletek implementálásához mindig egy programot kell megadni (de nem feltétlenül struktogram alakban).

# Tesztelési terv

Megvalósított műveletek tesztelése (fekete doboz tesztelés)

1. Különféle méretű halmazok létrehozása, feltöltése és kiírása.
   1. 0, 5, méretű
2. Halmazhoz elemek hozzáadása
   1. Már benne lévő elem
   2. Még nincs benne és kisebb *max*-nál
   3. Még nincs benne de nagyobb *max*-nál
3. A default ctor kipróbálása.
   1. Üres halmaz létrehozásával, max üres lesz és a vector is
   2. Több elemű lista létrehozásával a max a legnagyobb elem lesz a vector pedig N méretű lesz (capacity lehet több)
4. Halmazból elemek kivétele
   1. Halmazban szereplő elem kivétele
   2. Halmazban nem szereplő elem kivétele
5. A halmaz legnagyobb elemének lekérése
   1. ha nem üres a halmaz értéket kapunk
   2. ha üres a halmaz hiba kiíratása
6. Üres-e a halmaz
   1. Minden esetben használni kell az értéket, ugyanis ha nincs használva nem fordul le a program a compiler hibát dob
7. Halmaz kiíratása
   1. Ha üres
   2. Ha nem üres

Tesztesetek a kód alapján (fehér doboz tesztelés)

1. Üres illetve nem üres halmaz létrehozása
2. setMax függvény ellenőrzése
3. isContain függvény ellenőrzése