Programozás Beadandó

Tompa Gábor

27 Március 2018

1 Feladat

Valósítsa meg az egész számokat tartalmazó zsák típust! Ábrázolja a zsák elemeit (az előfordulás számukkal együtt) egy sorozatban! Implementálja a szokásos műveleteket (elem betevése, kivétele, üres-e a halmaz, egy elem hányszor van a zsákban), valamint két zsák szimmetrikus differenciáját (a közös elemek nem kerülnek be a szimmetrikus differenciába), továbbá egy zsák kiírását!

2 Zsak tipus

A zsák típus (Vagy más neven multiset) a klasszikus halmaz kibővitese azzal az engedménnyel hogy egy elemet többször is tartalmazhat(aminek számosságát taroljuk).A megvalósításnak része meg a halmaz hossza is.

3 Típus műveletek

3.1 Üres

Egy metódus ami ellenőrzi hogy a megadott zsák üres e

3.2 Szimmetrikus Differencia

Eg metódus ami egy másik Zsák típust értekül kapva visszaadja a két Zsák szimmetrikus differenciáját(mindegyik elemet csak egyszer rak bele a differenciába)

3.3 Betevés

Egy metódus ami beletesz egy új elemet a Zsákba

3.4 Kivetel

Egy elemet kivesz a zsákból,de teljesen(ha többször van benne akkor mindegyiket kiveszi)

3.5 Elem sokasága

egy metódus ami ellenőrzi hogy az adott elem hányszor található meg a zsákban

4 Reprezentáció

A Zsákot egy kettő egész számokból álló struktúrából (ami az elemet és a számosságát tartalmazza)alkotott egy dimenziós vektor alkotja valamint a hossz változó ami szinten egy természetes szám,a könnyebb debugolhatosagert,es a nagy zsákok lehetőségéért.

5 Implementáció

5.1 Szimmetrikus Differencia

A szimmetrikus differencia O(n^2)*alattimegvalósítása*

procedure SZIM_DIF(*other_bag*)

temp_container

benne_van

for i in *this.tarolo_meret* **do**

for j in *other_bag.tarolo_meret* **do**

if *this.tarolo_tomb*[*i*].*element* = *other_bag.tarolo_tomb*[*j*].*element* **then**

true ←benne_van **end if**

if *benne_van* = false **then**

this.tarolo_tomb[*i*].*element* ←PutIn(temp_container) **end if**

end procedure=0

5.2 Elem sokasaga

Egy elem sokaságának O(n) alatti megvalositasa

procedure ELEMENTCARDINALITY(*element*)

2: 0 ←cardinality **for** jother_bag.tarolo_meret **do**

4: *this.tarolo_tomb*[*i*].*element* = *element*

cardinality ←this.tarolo_tomb[i].cardinality

6:

8: **end procedure**

5.3 Betevés

Beletesz egy elemet a zsákba,ugy hogy lefoglal egy eggyel nagyobb átmeneti tömböt,amibe áttölti az adatokat,és utána törli a régit,felvesz egy eggyel nagyobb tarolót,amibe az átmeneti tömbből áttölti az elemeket

5.4 Kivetél

Hasonló a betevés művelethez,csak kiveszi az elemet és az elem nélkül maradékot tölti fel

5.5 Üres

Igazat ad vissza ha a hossz adattag 0 hamisat ha nem(mivel csak a betevés metóduson keresztül veszünk fel elemeket ahol gondoskodunk a méret adattag növeléséről,igy mindig helyes érteket fog visszaadni)

6 Osztaly

A Zsák osztályt egy Bag osztály valósítja meg ami tartalmaz egy kételemű struktúrából álló dinamikus tömböt és egy hossz érteket,valamint definiálja a szimmetrikus differencia függvényt,a PutIn (betevésre szolgáló),deleteElement(kivételre szolgáló),Üres() függvényt,Egy kiíró WriteOut() metódust ,egy ElementCardinality() metódust ami visszaadja egy elem számosságát,egy segédfüggvényt ami visszaadja a taroló tömböt,valamint újradefiniálja az értékadó operatort,és tartalmaz másoló konstruktort az üres és az elemeket magadon kívül.

Bag	
-	tarolo_meret : unsigned long long int
-	tarolo_tomb : element_and_cardinality[0...d]
+	Bag(int[],int length)
+	Bag()
+	Bag(Bag)
+	Szim_dif(Bag) : Bag
+	PutIn(int) : Bag
+	DeleteElement(int)
+	WriteOut()
+	ElementCardinality(int) : int
+	operator =(Bag) : Bag

A seged struktura:

element_and_cardinality	
-	element : int
-	cardinality : int

7 Tesztelési terv

1. Uj Zsákok létrehozása
   (a) Egy elemű zsák létrehozása
   (b) Két elemű Zsák létrehozása
   (c) Üres zsák létrehozása
2. Üres Zsákba új elem
3. 0 berakása a nullát mar tartalmazó zsákba
4. több elemű zsák egyeretelmu maxelemmel,Elemszám ellenőrzéssel
5. Sok betevési művelet
6. Sok torlesi művelet
7. Egymás után sokszor kivétel ugyanazzal az elemmel
8. Egyenlő szimdif(szimdif funkcionálási tesztelése)
9. Elem torlese meglevő elemekből
10. Másoló konstruktor
11. értékadás
12. értékadás egyenlően