



Belépő a tudás közösségébe

Informatika szakköri segédanyag



Verem típus

Menyhárt László

Szerkesztő: Abonyi-Tóth Andor, Zsakó László

A kiadvány „A felsőoktatásba bekerülést elősegítő készségfejlesztő és kommunikációs programok megvalósítása, valamint az MTMI szakok népszerűsítése a felsőoktatásban” (EFOP-3.4.4-16-2017-006) című pályázat keretében készült 2018-ban.



Eötvös Loránd Tudományegyetem
Informatikai Kar

SZÉCHENYI 2020



MAGYARORSZÁG
KORMÁNYA

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE

Ebben az anyagrészen egy speciális sorozat típusról olvashatunk.

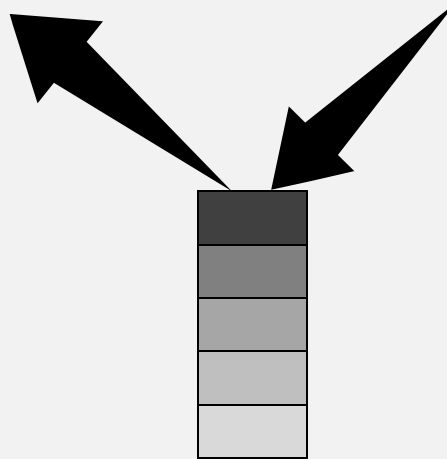
Ez az adatszerkezet nem csak bizonyos elemek tárolására lesz alkalmas, hanem műveletek is tartoznak hozzá, melyekkel lehet az elemeket elérni és így speciális logikát alkalmazhatunk rajta.

Az értékhalmoz: az alaphalmaz (amely az *Elemtípus* által van meghatározva) iteráltja („mely elemek lehetnek benne a halmazban”). Az *Elemtípus* általában valamely véges diszkrét típus lehet, legtöbbször még az elemszámát is korlátozzák (≤ 256). Ha nyelvi elemként nem létezik, akkor a megvalósításunkban lehet nagyobb elemszámú is.

Tegyük fel, hogy a probléma megismerése és a megoldás tervezése közben már tudjuk, hogy maximum mennyi adatot szeretnénk tárolni, így az egyszerűség kedvéért egy olyan statikus implementációt mutatunk be, ami tömböt használ az adatok tárolására.

Verem

A lényege, hogy az az adat, ami utoljára belekerül, azt vesszük ki először és ez igaz minden további elemre is.



Angolul last in, first out (LIFO) kifejezéssel szokták leírni. A hétköznapiakból is ismert lehet, például amikor egy irattartóra mindig a tetejére feldolgozandó iratokat és onnan is vesszük el a feladatokat. Az elnevezését egy szűk és mély tároló helyről kapta. Ugyanígy vannak speciális feladatok, amikor az adatokat is hasonlóképpen kell feldolgozni.

Műveletek

Üres (üres verem létrehozás eljárás), *Üres?* (logikai értékű függvény), *Verembe* (elem hozzávétele egy veremhez), *Veremből* (elem kivétele egy veremből), *Tető* (elem megtekintése a verem tetejéről).

A műveletek tulajdonságai közül a következők a legfontosabbak. Üres veremre a verembe rakás és az üresség vizsgálat kivételével minden művelet értelmetlen. A veremből kivétel a verembe utoljára tett elemet veszi ki, azaz a verem alkalmas egy sorozat megfordítására. A tetőn mindig a verembe utoljára betett (legfelül levő) elem látszik.

```
Verem (Elemtípus) = Rekord (elem: Tömb (1..MaxDb:Elemtípus); utolsó: Egész)
```

MaxDb elemszámú Elemtípus típusú elemet tárolunk egy tömbben. Egy segédváltozó mutatja, hogy melyik után kell felvenni új elemet, és ugyanezt lehet kivenni a veremből. Az elemek folytonosan szerepelnek az elsőtől az utolsóig.

Ebben az esetben az alábbi módon valósíthatjuk meg a műveleteket:

```
Eljárás Üres (Változó v: Verem(Elemtípus)) :  
    v.utolsó:=0  
Eljárás vége.
```

Műveletigény számítása: nem függ a **verem elemszámától**.

```
Függvény Üres? (Konstans v: Verem(Elemtípus)) : Logikai  
    Üres? := (v.utolsó=0)  
Függvény vége.
```

Műveletigény számítása: nem függ a **verem elemszámától**.

```
Eljárás Verembe (Változó v: Verem(Elemtípus),  
                Konstans e: elemtípus):  
    v.utolsó:=v.utolsó+1  
    v.elem[v.utolsó]:=e  
Eljárás vége.
```

Műveletigény számítása: nem függ a **verem elemszámától**.

```
Függvény Veremből (Változó v: Verem(Elemtípus)) : Elemtípus  
    u:=v.utolsó  
    v.utolsó:=v.utolsó-1  
    Veremből:=v.elem[u]  
Eljárás vége.
```

Műveletigény számítása: nem függ a **verem elemszámától**.

```
Függvény Tető (Konstans v: Verem(Elemtípus)) : Elemtípus  
    Tető:=v.elem[v.utolsó]  
Függvény vége.
```

Műveletigény számítása: nem függ a **verem elemszámától**.

Minta kódok.

C++ [cpp/feladat.cpp](#)

C# [cs/feladat.cs](#)

Java [java/feladat.java](#)

Pascal [pas/feladat.pas](#)

Python [py/feladat.py](#)



A megoldás alapvető problémája, hogy MaxDb-t előre kell ismernünk és természetesen bele kell férnie a memóriába.

Az így ábrázolt veremben lévő elemek típusára semmilyen megkötést nem kell tennünk, hiszen egy tömbben bármilyen elem elhelyezhető.

A verem implementációjánál még érdemes a hibákat is kezelni. Például hibát okoz, ha Üres veremből vennénk ki elemet, vagy teli verembe szeretnénk még tenni elemet.

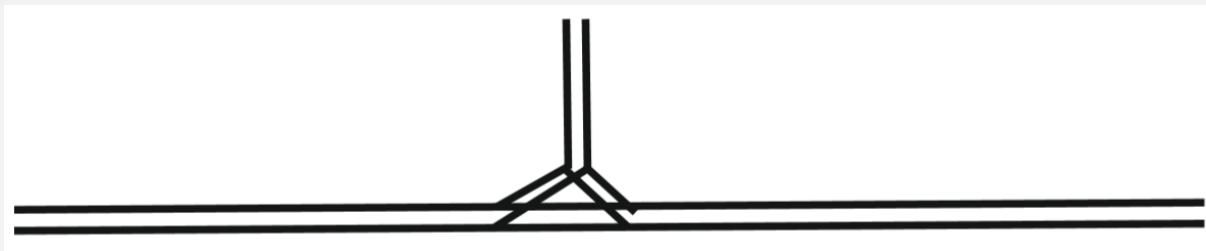
Feladatok halmazokra

a Nemes Tihamér NITV-ről és az Informatika OKTV-ről

1. feladat

Vasúti kocsik mellékvágányokon

Egy vasútállomáson a fő vágányról egy kitérő vágányt építettek, mindegyiken csak jobbról balra lehet haladni. A kitérőn akárhány kocsit elhelyezhetünk, a fő vágányon azonban nem állhat meg kocsi. Egy szerelvény érkezik jobbról, sorszámozott, összekevert kocsikkal. A kocsikat 1 és K közötti sorszámokkal azonosítjuk.



Készíts programot, amely megadja, hogy közülük maximum hány kocsit tudunk sorba rendezni a kitérő segítségével és elküldeni balra úgy, hogy a kitérőn végül nem marad kocsi!

A standard bemenet első sorában a kocsik száma van ($1 \leq K \leq 100\ 000$). A következő K sorban a kocsik sorszámjai következnek az érkező szerelvényen belüli sorrendben ($1 \leq S_i \leq K$).

A standard kimenet egyetlen sorába azt kell kiírni, hogy maximum mennyi kocsit lehet így sorba rendezni!

Példa:

Bemenet

6
5
4
2
3
6
1

Kimenet

5

Magyarázat: a kocsik 2,3,4,5,6 sorrendben mehetnek ki balra.

Megoldás:

```

Olvas(k); ut:=0; jo:=igaz; Ures(v); i:=1
Ciklus amíg i≤k és jo
  Olvas(szam)
  Ha szam<út akkor jo:=hamis
  különben ha Üres?(v)
    akkor Verembe(v,szam)
  különben ha szam<Tető(v)
    akkor Verembe(v,szam)
  különben Ciklus amíg nem Ures?(v) és szam>Tető(v)
    ut:=Veremből(v)
    Ciklus vége
    Verembe(v,szam)
  Elágazás vége
  i:=i+1
Ciklus vége
Ha jo akkor Kiír(k) különben Kiír(i-2)
Eljárás vége.

```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Sorozatok
Feladat	79. Vasúti kocsik kitérő vágányokon

2. feladat

A veremautomata olyan gép, amely a bemenetként kapott számsorozaton az alábbi módon működik. Sorban balról-jobbra egyesével olvassa a számsorozatot és vagy a sorozat aktuális elemével, vagy a verem tetején lévő elemmel végezhet műveletet. Egy lépésben az alábbi három művelet valamelyikét hajthatja végre:

1. A bemenet aktuális elemét kiírja a kimenetre.
2. A bemenet aktuális elemét beteszi a verembe az ott lévő sorozat elé.
3. A verem tetején lévő (a sorozatban első) elemet kiveszi a veremből és kiírja a kimenetre.

Kezdetben a verem üres. Feladatunkban a veremautomatát arra akarjuk használni, hogy bemenetként kap egy számsorozatot, amely az $1, 2, \dots, N$ számokat tartalmazza tetszőleges sorrendben, és a kimenetre írja ki az $1, 2, \dots, M$ számsorozatot, a lehető legnagyobb M -ig. (A kimenetben minden számnak szerepelnie kell M -ig és sorrendben kell lenniük!)

Készíts programot, amely kiszámítja, hogy melyik az a legnagyobb M érték, amelyre a veremautomata kimenete az $1, 2, \dots, M$ lehet!

A standard bemenet első sorában a bemeneti sorozat elemszáma van ($1 \leq N \leq 10\,000$), a második sor pedig az $1, 2, \dots, N$ számok valamely permutációját tartalmazza, a számokat egy-egy szóközzel elválasztva.

A standard kimenet első és egyetlen sorába azt a legnagyobb M számot kell írni, amelyre a veremautomata kimenete az $1, 2, \dots, M$ sorozat lehet!

Példa:**Bemenet**

10
3 2 1 5 4 6 9 7 10 8

Kimenet

8

Megoldás:

```

Olvas(N)
Ciklus i:=1-től N-ig
  Olvas(A[i])
Ciklus vége
Sveg:=0; i:=1; Üres(v)
Ciklus amíg i≤N
  Ha A[i]=Sveg+1 akkor Inc(SVeg); Inc(i)
  különb ha Tető(v)=Sveg+1
    akkor Inc(SVeg); Veremből(v)
  különb Verembe(v,A[i]); Inc(i)
Elágazás vége
Ciklus vége
Ciklus amíg Tető(v)≠0 és Tető(v)=Sveg+1
  Inc(SVeg); Veremből(v)
Ciklus vége
Kiír(SVeg)

```

Eljárás vége.

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Sorozat
Feladat	38. Veremautomata

3. feladat

Egy síugró versenyen minden versenyző akkor indulhat, ha az összes korábban indult versenyző befejezte már a versenyzését. Egy versenyző akkor győz le egy másikat, ha ugrásával magasabb pontszámot ér el.

Készíts programot, amely megadja minden indulóra, hogy az előtte indulók közül ki volt a hozzá legközelebb indult, általa legyőzött versenyző, valamint az utána indulók közül a hozzá legközelebb indult olyan, aki legyőzte őt!

A standard bemenet első sorában a versenyzők száma van ($1 \leq N \leq 100\,000$). A következő N sor mindegyike egy-egy versenyző pontszámát tartalmazza ($1 \leq P_i \leq 100\,000$), az indulás sorrendjében.

A standard kimenet i . sorába az i . induló versenyző előtteiek közül az utolsónak induló, általa legyőzött versenyző sorszáma, valamint az utána indultak közül a leghamarabb induló őt legyőző versenyző sorszáma kerüljön! Ha valaki senki őt megelőzőt nem győzött le, akkor a sor első száma -1 legyen! Ha valakit senki őt követőt nem győzött le, akkor a második szám legyen -1 !

Példa:**Bemenet**

7
3
2
5
6
1
4
3

Kimenet

-1 3
-1 3
2 4
3 -1
-1 6
5 -1
5 -1

Megoldas:

```

Olvas(n);
Üres(v); pont[0]:=-1
Ciklus i:=1-től n-ig
  Olvas(pont[i])
  Ciklus amíg pont[Tető(v)]≥pont[i]
    Veremből(v)
  Ciklus vége
  Ha Tető(v)=0 akkor elo[i]:=-1
  különben elo[i]:=v[teteje]
  Verembe(v,i)
Ciklus vége
pont[0]:=maxint; Üres(v)
Ciklus i:=n-től 1-ig -1-esével
  Ciklus amíg pont[Tető(v)]≤pont[i]
    Veremből(v)
  Ciklus vége
  Ha Tető(v)=0 akkor kov[i]:=-1
  különben kov[i]:=v[teteje]
  Verembe(v,i)
Ciklus vége
Ciklus i:=1-től n-ig
  Kiír(elo[i], ' ', kov[i])
Ciklus vége

```

Eljárás vége.

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Sorozat
Feladat	61. Legyőzött és legyőző versenyző

4. feladat

Egy sűrű versenyen minden versenyző akkor indulhat, ha az összes korábban indult versenyző befejezte már a versenyzését. Egy versenyző akkor győz le egy másikat, ha ugrásával magasabb pontszámot ér el.

Készíts programot, amely megadja minden indulóra, hogy az előtte és utána indulók közül ki volt a hozzá legközelebbi indult, általa legyőzött versenyző, valamint az előtte és utána indulók közül a hozzá legközelebbi indult olyan, aki legyőzte őt!

A standard bemenet első sorában a versenyzők száma van ($1 \leq N \leq 100\,000$). A következő N sor mindegyike egy-egy versenyző pontszámát tartalmazza ($1 \leq P_i \leq 100\,000$), az indulás sorrendjében.

A standard kimenet i . sorába az i . induló versenyző előtteiek közül az utolsónak induló, általa legyőzött versenyző sorszáma, valamint őt legyőző versenyző sorszáma kerüljön! Ha valaki senki őt megelőzőt nem győzött le, akkor a sor első száma -1 legyen; ha senki megelőző nem győzte le őt, akkor a második szám legyen -1 ! Ha valakit senki őt követőt nem győzött le, akkor a második szám legyen -1 ! A következő N sorban ugyanez szerepeljen az adott versenyzők utáni versenyzőkkel!

Példa:

Bemenet	Kimenet
7	-1 1
3	-1 1
2	2 -1
5	3 -1
6	-1 4
1	5 4
4	5 6
3	2 3
	5 3
	5 4
	5 -1
	-1 6
	7 -1
	-1 -1

Megoldás:

```

Olvas(n)
Üres(v); pont[0]:=-1
Üres(v2) pont[n+1]:=maxint
Ciklus i:=1-től n-ig
  Olvas(pont[i])
  Ciklus amíg pont[tető(v)]≥pont[i]
    Veremből(v)
  Ciklus vége
  Ha Tető(v)=0 akkor elok[i]:=-1
  különben elok[i]:=v[teteje]
  Verembe(v,i)
  Ciklus amíg pont[Tető(v2)]≤pont[i]
    Veremből(v2)
  Ciklus vége
  Ha Tető(v2)=0 akkor elon[i]:=-1
  különben elon[i]:=Tető(v2)
  Verembe(v2,i)
Ciklus vége

```



```

Üres(v); Üres(v2)
Ciklus i:=n-től 1-ig -1-esével
  Ciklus amíg pont[Tető(v)]≤pont[i]
    dec(teteje)
  Ciklus vége
  Ha Tető(v)=0 akkor kovn[i]:=-1
  különben kovn[i]:=Tető(v)
  Verembe(v,i)
  Ciklus amíg pont[Tető(v2)]≥pont[i]
    Veremből(v2)
  Ciklus vége
  Ha teteje2=0 akkor kovk[i]:=-1
  különben kovk[i]:=Tető(v2)
  Verembe(v2,i)
Ciklus vége
Ciklus i:=1-től n-ig
  Kiír(elok[i], ' ',elon[i])
Ciklus vége
Ciklus i:=1-től n-ig
  Kiír(kovk[i], ' ',kovn[i])
Ciklus vége

```

Eljárás vége.

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Sorozat
Feladat	62. Legyőzött és legyőző versenyzők

5. feladat

Egy kifejezés (,) és [,] zárójelpárokat tartalmaz.

Készíts programot, amely ellenőrzi a helyességét!

A standard bemenet első sorában a zárójeles kifejezés karaktereinek száma van ($1 \leq N \leq 100000$). A következő sorban egy N hosszú kifejezés, amit ellenőrizni kell.

A standard kimenet első sorába 1-est vagy 0-t kell írni úgy, hogy az a bemenetben szereplő kifejezés helyességét mutassa! 1 jelenti azt, hogy helyes a kifejezés, míg a 0 azt, hogy hibás.

Példák:**Bemenet**

```
5
x(42)
```

Magyarázat: helyes

Kimenet

```
1
```

Bemenet

```
6
x([y])
```

Magyarázat: hibás párosítás

Kimenet

```
0
```

Bemenet

3 x(a

Kimenet

0

Magyarázat: nyitó zárójelnek nincs csukó párja

Bemenet

4 x)a(

Kimenet

0

Magyarázat: csukó zárójelnek nincs nyitó párja

Megoldás:

```

Olvas(n)
i:=1; h:=igaz; Üres(V)
Ciklus amíg i≤n és h
  Olvas(kar)
  Ha kar=' (' vagy kar=' [' akkor
    Verembe(V, kar)
  különben ha kar=')' vagy kar=']' akkor
    Veremből(V, s)
    h:=nem Hibás-e(V) és ( kar=')' és s=' ('
      vagy kar=']' és s=' [' )
  Elágazás vége
  i:=i+1
Ciklus vége
Ha h és üres?(V) akkor
  Kiír(1)
különben
  Kiír(0)
Elágazás vége

```

Eljárás vége.

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	
Téma	
Feladat	