

Algoritmusok és adatstruktúrák (1.előadás)

Az elsajátítandó ismeretanyag:

Az algoritmus fogalma. Az algoritmusok elemzésének és tervezésének alapfogalmai. Algoritmusok ábrázolásának módjai. Egyszerű algoritmusok. Rendezési algoritmusok. Keresési algoritmusok. Rekurzivitás, rekurzív algoritmusok. Algoritmusok hatékonyságának összehasonlítása. Aszimptotikus viselkedések és jelölések. Adatszerkezetek ábrázolásai és műveleteik: verem és sorok, láncolt listák, kupacok. Fák ábrázolása, bináris kereső fák és műveleteik, gráfok ábrázolásai, gráfelméleti algoritmusok.

Félévközi tanulmányi követelmények:

Egy 40 pontos elméleti zárthelyi dolgozat a 11. héten és két 30 pontos zárthelyi dolgozat a 6. és a 12. héten. Javítási lehetőség a 13. héten. A **gyakorlati jegy (GINFBAN-ALGOADAT-2 és GPTIFKN-ALGOADAT-2 kurzusok)** vagy a vizsgára bocsátás (**GINFBAN-ALGOADAT-1 és GPTIFKN-ALGOADAT-1 kurzusok**) feltételei: Az elméleti dolgozatban legalább 20 pont és a gyakorlati dolgozatokban legalább 30 pont elérése.

A gyakorlati jegy megállapítása (**GINFBAN-ALGOADAT-2 és GPTIFKN-ALGOADAT-2 kurzusok**) az elért összpontszám alapján történik a TVSz-ben megadott értékhatárok szerint.

A vizsgás kurzusok esetén (**GINFBAN-ALGOADAT-1 és GPTIFKN-ALGOADAT-1 kurzusok**) megajánlott jegy adható a TVSz-ben leírtak szerint.

Algoritmusok és adatstruktúrák (1.előadás)

Algoritmus: olyan pontos előírás , amely megadja , hogy egy feladat megoldásához milyen műveleteket, milyen sorrendben kell elvégezni.

Elvárások:

- **Algoritmus végeessége** : csak véges sok lépésből állhat
- **Egyértelműség**: nem lehetnek az algoritmusnak kétértelmű részei
- **Teljesség** : a feladathoz szükséges összes lépést tartalmazni kell

Az algoritmus nem program, csak megoldási vázlat

- Algoritmus fejlesztési módja: lépésenkénti finomítás elve
- Algoritmus fő részei:
 - A bemenő adatok bevitele
 - A szükséges műveletek elvégzése
 - Az eredmények megjelenítése, rögzítése, kiírása

Algoritmusok és adatstruktúrák (1.előadás)

Programozási feladatok megoldásának lépései

1. A feladat meghatározása

- egyértelmű, rövid, tömör, pontos

2. A feladat algoritmusának elkészítése

3. Kódolás

- a kész algoritmus valamilyen konkrét programnyelvre való lefordítása

Program: a gép számára érthető formájú meghatározott sorrendű, véges, gépi utasítássorozat.

4. Tesztelés

5. Dokumentálás (felhasználói és fejlesztői)

Algoritmusok és adatstruktúrák (1.előadás)

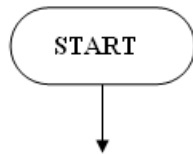
Algoritmus leíró eszközök

- Céljuk: a megoldás menetének géptől független, szemléletes, a logikai gondolatmenetet, a szerkezeti egységeket világosan tükröző leírása
- Legismertebb algoritmus leíró eszközök:
 - a) folyamatábra
 - b) mondatszerű leírások

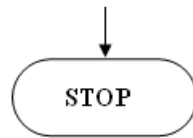
Algoritmusok és adatstruktúrák (1.előadás)

Folyamatábrák

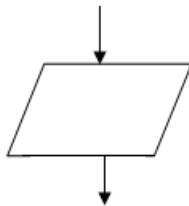
Algoritmus kezdete:



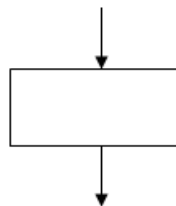
Algoritmus vége:



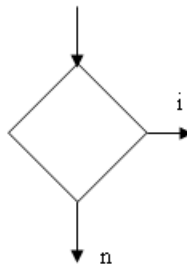
Beolvasás, kiírás:



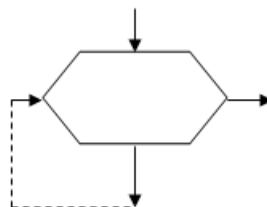
Művelet végrehajtása:



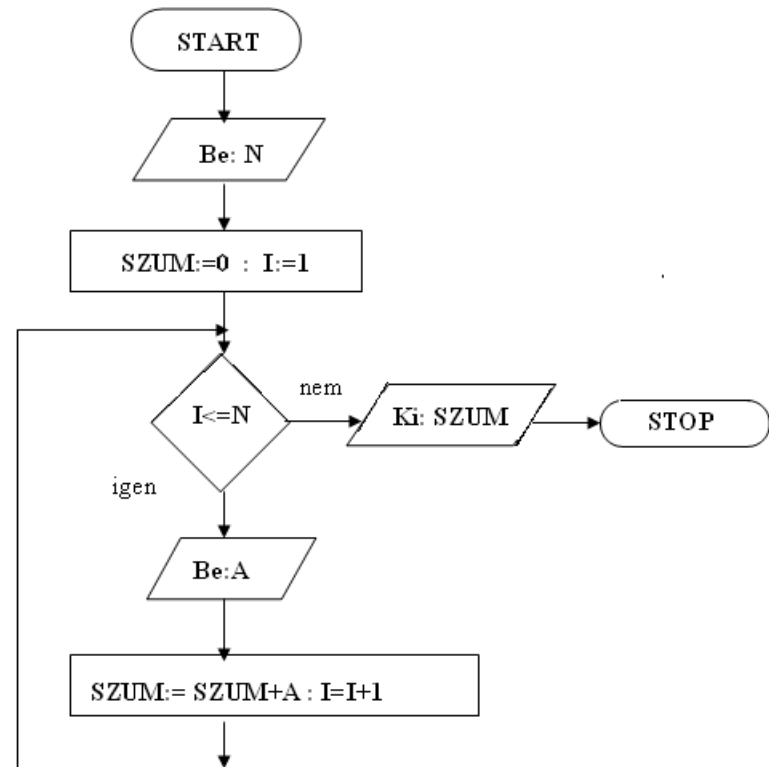
Elágazás (feltétel alapján egyik vagy másik irányban folytatódik):



Számláló ciklus jelölésére használt szimbólum:



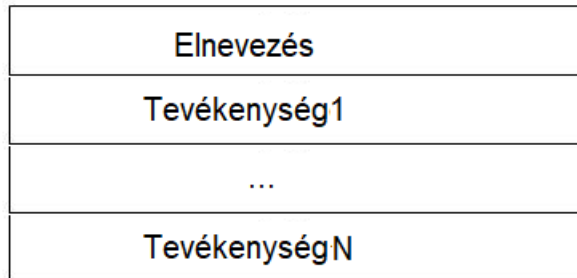
Példa: Kérjünk be a billentyűzetről N (>0, egész) számot és számoljuk ki az összegüket!



Algoritmusok és adatstruktúrák (1.előadás)

Strukturogramok

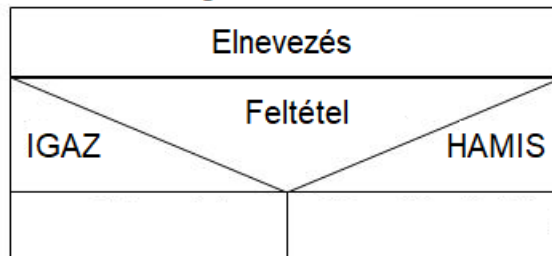
Blokk leírása:



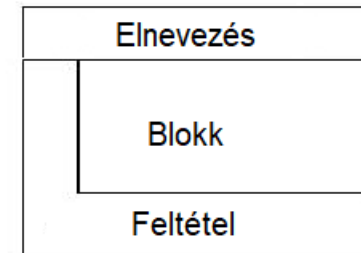
Elöl tesztelő ciklus:



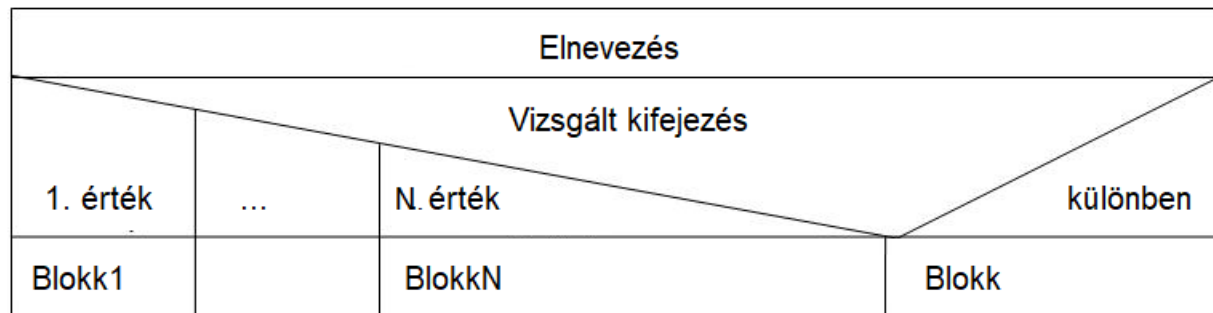
Feltételes elágazás:



Hátul tesztelő ciklus:



Több ágú elágazás:



Algoritmusok és adatstruktúrák (1.előadás)

Mondatszerű leírások

- A **mondatok sorozatával** írjuk le a feladat megoldását, s a mondatokat sorszámozzuk.
A program egyes részeiben , ha el kell térni a szekvenciális végrehajtástól a megfelelő sorszámba hivatkozhatunk.
- A mondatszerű leírások szemléletesebb módja az amikor mondatok helyett csak ún. **mondatszerű szerkezetek** egymásutánjával írjuk le a feladat megoldását. Ennél a leírási módnál az algoritmus szerkezeti egységeit bekezdéses struktúra segítségével szemléltetjük.

Algoritmusok és adatstruktúrák (1.előadás)

Mondatszerű szerkezetek:

Beolvasó és kiíró utasítás

Formája :

Be:	//
változók felsorolása	az adatokkal szemben támasztott követelmények

Ki:	//
kifejezések, változók	a kiírás formájára vonatkozó követelmények

Értékadó utasítás

Formája:

.....	:=
változó		kifejezés, változó

Az utasítás hatására a bal oldalon álló változó felveszi a jobb oldal értékét.

Algoritmusok és adatstruktúrák (1.előadás)

Elágazások (feltételes utasítások)

Formája :

A, **Ha** (.....) **akkor**
 valamilyen feltétel utasítás(ok)

Elágazás vége

B, **Ha** (.....) **akkor**
 valamilyen feltétel utasítás(ok)1

különben
 utasítás(ok)2

Elágazás vége

C, **Elágazás****esetén**
 feltétel 1 utasítás(ok) 1

 **esetén**
 feltétel 2 utasítás(ok) 2

.

.

 **esetén**
 feltétel n utasítás(ok) n

egyéb esetben
 utasítás(ok) n+1

Elágazás vége

Algoritmusok és adatstruktúrák (1.előadás)

Egyéb jelölések:

// kiegészítések , megjegyzések

: ha egy soron belül több utasítást használunk

tomb[] - tömbök jelölése (**tomb [i]** a tömb i indexű eleme)

tomb[1..N] – n elemű tömb 1, 2, ... N indexekkel

tomb[0..N-1] – n elemű tömb 0, 1, ... , N-1 indexekkel (C-like)

Ha teljes programot írunk:

Program:

 program utasításai

Program vége

Ha eljárást(függvényt írunk):

eljárásnév.(.....)

 paraméterek

eljárás utasításai

eljárás vége

Algoritmusok és adatstruktúrák (1.előadás)

FELADAT : Kérjünk be a billentyűzetről N (>0 , egész) számot és számoljuk ki az összegüket!

Program

Be: N // $N > 0$ egész

Összeg:=0

Ciklus $i:=1$ - től N -ig

Be: A

Összeg:= Összeg + A

Ciklus vége.

Ki: Összeg

Program vége.

Algoritmusok és adatstruktúrák (1.előadás)

Összegzés

Bemenet: $A[0..N-1]$

Kimenet: Összeg

Algoritmus:

 Összeg:=0

 Ciklus $i := 0$ -tól $N-1$ -ig

 Összeg := Összeg + $A[i]$

 Ciklus vége.

Algoritmus vége.

Algoritmusok és adatstruktúrák (1.előadás)

Pl. A $T[0..11]$ tömb tartalmazza egy gyárnak 2021-ben havonta termelt mennyiségeket.

Mennyi volt a gyár 2021-es össztermelése?

Mennyi volt a gyár 2021-es átlagos havi termelése?

Bemenet: $T[0..11]$

Kimenet: Össz, Átlag

Algoritmus:

Össz:=0

Ciklus $i:=0$ -tól 11-ig

Össz := Össz + $T[i]$

Ciklus vége

Atl:=Össz/12

Algoritmus vége.

Algoritmusok és adatstruktúrák (1.előadás)

Eldöntés

Adott egy $N > 0$ elemű sorozat, s egy a sorozat tagjain értelmezett T tulajdonság. Olyan algoritmust kell írni amely eldönti : **van-e** a sorozat tagjai közt legalább egy T tulajdonságú elem.

Bemenet: $A[0..N-1]$, T tulajdonság

Kimenet: Eredmény // 'VAN T tulajdonságú' vagy 'NINCS T tulajdonságú'

Algoritmus:

$i := 0$

 Ciklus amíg $i < N$ és $A[i]$ nem T tulajdonságú

$i := i + 1$

 Ciklus vége

 Ha($i < N$) akkor Eredmény $:=$ ' VAN T tulajdonságú elem'

 különben Eredmény $:=$ ' NINCS T tulajdonságú elem'

 Elágazás vége

Algoritmus vége.

Algoritmusok és adatstruktúrák (1.előadás)

Kiválasztás

Adott egy $N > 0$ elemű sorozat, és egy a sorozat tagjain értelmezett T tulajdonság. **Tudjuk, hogy** a sorozat tagjai közt **van T tulajdonságú**. A feladat megadni az első **T tulajdonságú** tag sorszámát.

Bemenet: $A[0..N-1]$, T tulajdonság, tudjuk, hogy van T tulajdonságú elem

Kimenet: Sorsz // a T tulajdonságú elem sorszáma

Algoritmus :

$i := 0$

 Ciklus amíg $A[i]$ nem T tulajdonságú

$i := i + 1$

 Ciklus vége.

 Sorsz := i // Sorsz := $i + 1$, ha az első elem sorszáma 1

Algoritmus vége.

Algoritmusok és adatstruktúrák (1.előadás)

Keresés

Adott egy $N > 0$ elemű sorozat, s egy a sorozat tagjain értelmezett T tulajdonság. Olyan algoritmust kell írni amely eldönti, hogy van-e T tulajdonságú elem a sorozatban és ha van hányadik. Ez a tétel gyakorlatilag az eldöntés és kiválasztás összegzése.

Lineáris keresés

Bemenet: $A[0..N-1]$, T tulajdonság.

Kimenet: Sorsz // a T tulajdonságú elem sorszáma – ha van, -1 – ha nincs

Algoritmus:

$i := 0$

 Ciklus amíg $i < N$ és $A[i]$ nem T tulaj.

$i := i + 1$

 Ciklus vége

 Ha $(i < N)$ akkor Sorsz := i // Sorsz := $i + 1$, ha az első elem sorszáma 1
 különben Sorsz := -1

 Elágazás vége

Algoritmus vége.

Algoritmusok és adatstruktúrák (1.előadás)

Logaritmikus keresés

Adott egy $N > 0$ elemű RENDEZETT sorozat és egy keresett elem (Z). A sorozat elemeit az $A[0..N-1]$ tömbben tároljuk. Olyan algoritmust kell írni amely eldönti, hogy van-e (Z) elem a sorozatban és ha van hányadik.

Bemenet: $A[0..N-1]$ A rendezett sorozat, Z a keresett elem

Kimenet: Sorsz // a Z elem sorszáma ha van, -1 ha nincs

Algoritmus:

$AH := 0$: $FH := N-1$ //a sorozat alsó határa-AH felső határa-FH

 Ciklus

$i := EGÉSZRÉSZ((AH+FH)/2)$ // az egészrész függvény egy tört egészrészét adja

 Ha($A[i] < Z$) akkor $AH := i+1$

 Ha($A[i] > Z$) akkor $FH := i-1$

 amíg $AH \leq FH$ és $A[i] \neq Z$

 Ha($AH \leq FH$) akkor Sorsz := i // Sorsz:= $i+1$, ha az első elem sorszáma 1

 különben Sorsz := -1

 Elágazás vége

Algoritmus vége.

A **logaritmikus** elnevezés onnan származik, hogy ezzel a módszerrel az elem megkereséséhez szükséges lépések száma maximum **$\log_2 N$**

Algoritmusok és adatstruktúrák (1.előadás)

Pl. Az N elemű osztálynévsorban van-e "Nagy" nevű, és ha van hányadik? A névsort a NEV[0..N-1] tömbben tároljuk.

Bemenet: NEV[0..N-1]

Kimenet: Sorsz

Algoritmus

AH:=0 : FH:=N-1

Ciklus

i:= EGÉSZRÉSZ((AH+FH)/2)

Ha(NEV[i]<"Nagy" akkor AH := i+1

Ha(NEV[i]>"Nagy" akkor FH := i-1

amíg AH<=FH és NEV[i] ≠ "Nagy"

Ha(AH<=FH) akkor Sorsz:=i+1

különben Sorsz := -1

Elágazás vége

Algoritmus vége.