Informatika 1 – 2011

Második előadás
Adattípusok, vezérlési szerkezetek

Szabó Adrienn

2011. szeptember 14.

Tartalom

A kérdőív eredményei

Köszönjük hogy kitöltöttétek!

- 49% érettségizett (középszinten, jó jegyekkel)
- 30%-nak van ECDL vizsgája
- 20% tanult már programozni (főleg Logo, Pascal, C#)
- Elvárás: megtanulni programozni

Algoritmusok

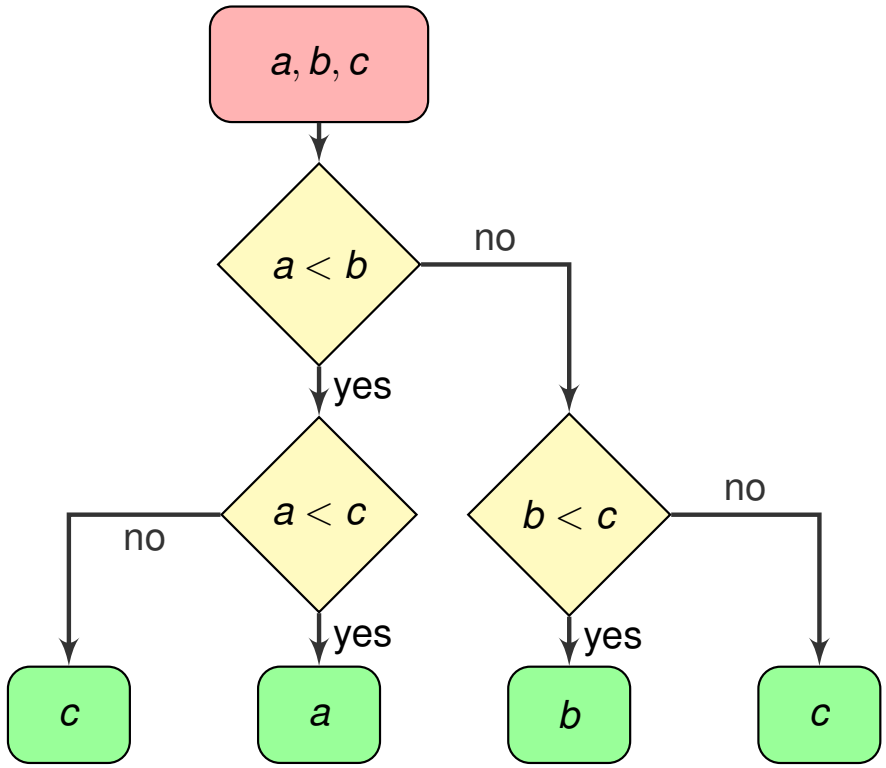
Algoritmuson vagy eljáráson olyan megengedett lépésekből álló módszert, utasítás(sorozato)t, részletes útmutatást, receptet értünk, amely valamely felmerült probléma megoldására alkalmas.

Az algoritmust leírhatjuk *pszeudokóddal* vagy *folyamatábrával*, illetve implementálhatjuk (megvalósíthatjuk) egy konkrét programnyelven, hogy működő programot kapjunk.

Algoritmus példa

Válasszuk ki a, b, c számok közül a legkisebbet!

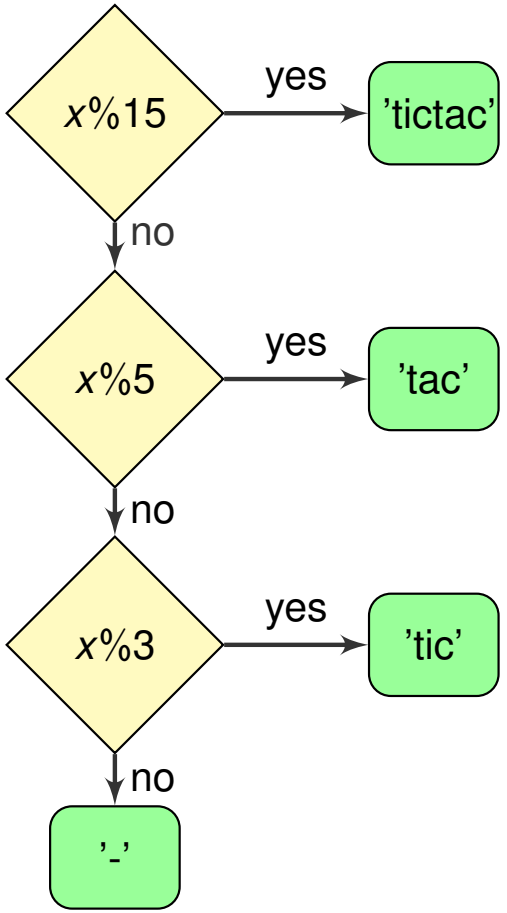
```
IF a < b:
    IF a < c:
        RETURN a
    ELSE:
        RETURN c
ELSE:
    IF b < c:
        RETURN b
    ELSE:
        RETURN c
```



Feltételes utasítás

Egy feltételes utasítás így néz ki Sage-ben, ill. Python-ban:
(Az `elif`, `else` részek el is hagyhatók)

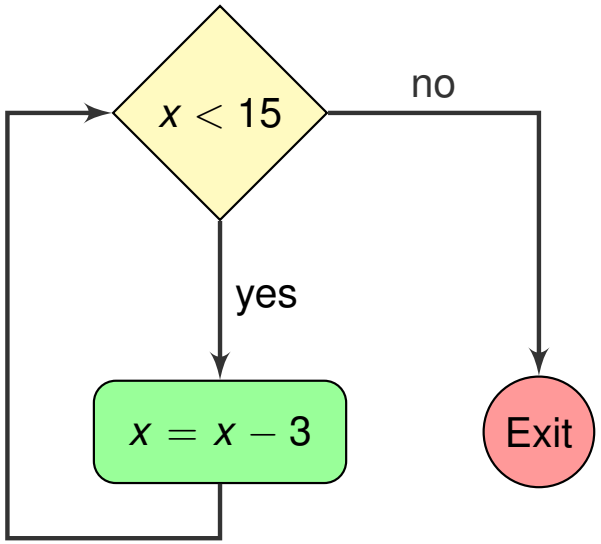
```
if x%15 == 0:
    return 'tictac'
elif x%5 == 0:
    return 'tic'
elif x%3 == 0:
    return 'tac'
else:
    return '-'
```



While ciklus

A `while`-ciklus belseje addig fog ciklikusan újra és újra végrehajtódni, amíg a feltétel része teljesül. Sage-ben:

```
while x > 15:
    x = x-3
    print x
```



Figyelni kell, hogy ne kerüljön a program végtelen ciklusba!

For ciklus

A `for` ciklus valamilyen lista vagy szekvencia elemein megy végig egyesével.

```
for x in range(4):
    print str(x) + '.  elemnél járunk'
```

A fenti kód kimenete:

```
0.  elemnél járunk
1.  elemnél járunk
2.  elemnél járunk
3.  elemnél járunk
```

Egyszerű adattípusok

A Sage (és kb. egyben a Python) beépített egyszerű típusai:

Típus	*Leírás*	*Példa*
`None`	„semmi”, null típus	`None`
`int`	egész szám (32-bites)	`6`
`long`	hosszú egész (tetszőlegesen hosszú)	`2354099L`
`float`	lebegőpontos szám (törtek)	`3.75`
`complex`	komplex szám	`3-2*i`

Összetett típusok

A Sage (és kb. egyben a Python) beépített összetett típusai:

Típus	*Leírás*	*Példa*
`str`	karakterlánc (string)	`'alma'`
`list`	lista	`[2, y, 'bb']`
`tuple`	nem írható lista	`(2, x, 'aa')`
`set`	halmaz	`set(['a', 'c'])`
`dict`	szótár (kulcs: érték)	`dict({'one':1, 'two':2})`

Listák – bevezető

- Listát definiálhatunk úgy, hogy megadjuk az elemeit (akármilyen típusúak lehetnek) szögletes zárójelek között:

```
sage: L1 = [pi, 'abc', 35, pi, 12]
```

- Hivatkozhatunk a lista egy elemére (itt a harmadikra):

```
sage: L1[2]
35
```

- Kiválaszthatunk egy rész-listát:

```
sage: L1[1:4]
['abc', 35, pi]
```

- Figyeljünk arra, hogy 0-tól kezdődik az elemek számozása!

Listák – bevezető 2

- Létrehozhatunk egy listát a `range` függvény segítségével:

```
sage: L2 = range(5)
sage: L2
[0, 1, 2, 3, 4]
```

- Lekérdezhetjük a lista hosszát (hány eleme van):

```
sage: len(L2)
5
```

- A listának bármely elemét felülírhatjuk:

```
sage: L2[1] = 'egy'
sage: L2
[0, 'egy', 2, 3, 4]
```

A `tuple` típus

A `tuple` sokmindenben hasonlít a listához, de az elemei utólag nem módosíthatóak.

- Kerek zárójellel definiáljuk:

```
sage: T = (pi, 'abc', 35, pi, 12)
```

- Hivatkozhatunk a tuple egy elemére:

```
sage: T[2]
35
```

- Ha meg akarnánk változtatni a tuple egy elemét:

```
sage: T[1]='s'
TypeError: 'tuple' object does not
support item assignment
```

A halmaz típus

A `set` típus megfelel a matematikai halmaz fogalomnak, nem rendezett elemek gyűjteménye.

- Definiálható a `set` kulcsszóval és egy lista megadásával:

```
sage: S = set([pi, 'abc', 35, pi, 12])
```

- Megkérdezhetjük, hogy valami benne van-e a halmazban:

```
sage: 35 in S
True
```

- Lekérdezhetjük a halmaz méretét (hány eleme van):

```
sage: len(S)
```

- Törlés a halmazból:

```
sage: S.remove(pi)
```

A szótár típus

A szótár arra való, hogy kulcs-érték párokat egymáshoz rendelhessünk.

- Definiálható a `dict` kulcsszóval a következő módon:

```
D = dict('one':1, 'two':2, 'three':3)
```

- Kulcshoz tartozó érték lekérdezése:

```
sage:  x = D['one']
```

- Megkérdezhetjük, hogy valami benne van-e a szótárban):

```
sage:  'two' in D
True
```

- Az értékek között is kereshetünk a `values` függvénnyel:

```
sage:  2 in D.values()
True
```

Példák

Nézd meg a wikin az előadáshoz tartozó Sage-worksheet-et!

További olvasnivalók

Hasznos olvasnivalók, néhol a tananyagon túl:

- Programozás bevezető (angolul):
  `http://johnstachurski.net/book/sample2.pdf`
- Sage Tutorial:
  `http://www.sagemath.org/pdf/SageTutorial.pdf`
- Algoritmus (Wikipédia oldal):
  `http://hu.wikipedia.org/wiki/Algoritmus`
- Folyamatábra (angol Wikipédia oldal):
  `http://en.wikipedia.org/wiki/Flow_chart`