

Diszkrét matematika

1. előadás

MÁRTON Gyöngyvér
<https://ms.sapientia.ro/~mgyongyi/>
mgyongyi@ms.sapientia.ro

Sapientia EMTE,
Matematika-Informatika Tanszék
Marosvásárhely, Románia

2024, őszi félév



Áttekintő - diszkrét matematika

- számok, számtartományok

- gyorsítványozás algoritmusai,
- algoritmusok hatékonysága,
- rekurzió,
- kiválogatás, minimum érték meghatározása,
- legnagyobb közös osztó, legkisebb közös többszörös,
- a Farey sorozat elemei,
- lánc törtek,
- irracionális számok számjegyei,
- a Fibonacci számsorozat,
- polinom helyettesítési értéke,
- fraktálok

Áttekintő - diszkrét matematika

- számrendszerek és kódolási technikák
 - természetes szám alapú számrendszerek,
 - számrendszerek közötti kapcsolat,
 - vegyes alapú számrendszerek,
 - bitműveletek,
 - az ASCII kódolás
 - az Unicode szabvány és az utf-8 kódolás
 - a base64 kódolás

Áttekintő - diszkrét matematika

- számelméleti alapfogalmak
 - prímszámok, prímtesztelő, prímfaktorizációs algoritmusok
 - kongruenciák, maradékosztályok, maradékrendszerek,
 - hatványok és generátor elemek,
 - az eukleidészi algoritmus és változatai
 - lineáris kongruenciák,
 - kongruencia rendszerek, a Kínai maradéktétel,
 - alkalmazások: a textbook-RSA publikus kulcsú kriptorendszer, a Diffie-Hellmann kulcscsere,
 - diofantoszi egyenletek,

Áttekintő

- **álvéletlen-szám generátorok**
 - lineáris kongruencián alapuló generátor,
 - a közép-négyzet módszer,
 - a Python álvéletlen számgenerátora
- **kombinatorika**
 - lexicografikus sorrend, permutációk, kombinációk, variációk

Követelmények, osztályozás

- egy egyetemi félév 14 hétből áll, Diszkrét matematikából egy héten 2 óra előadás, 2 óra gyakorlat van
- vizsgajegy: $\text{pluszPont} + (\text{felmérő1} + \text{felmérő2})/2$,
- 1 pluszPont-ot előadásokon feltett 3 kérdés megválaszolásával lehet szerezni,
- 1 pluszPont értéke 1 jegy
- maximálisan 3 pluszPont szerezhető
- a pluszPontok számától függetlenül az átmenő jegyhez a két felmérő átlaga mindig **nagyobb** kell legyen mint 4.5
- aki nem szerez előadáson pluszPont-ot annak a **maximális jegye 9**
- aki több pluszPont-ot is szerez annak a végső jegyéhez adódik hozzá a jegy, de csak akkor ha a két felmérő átlaga nagyobb mint 4.5
- a félév alatt maximum **3 pluszPont szerezhető**
- a felmérő1 és felmérő2 egy-egy feladatsorból fog állni, amelyeket számítógépen kell leprogramozni Python programozási nyelvben,
- a felmérő1 értéke nem kell nagyobb kell legyen, mint 4.5, és csak akkor kerül beszámításra ha a diák akarja
- a **felmérő2** értéke kötelező módon nagyobb kell legyen, mint 4.5

Követelmények, osztályozás

- a felmérő1-re a **félév közepén**, a felmérő2-re az **utolsó oktatási héten** kerül sor,
- a laborórákon való **jelenlét kötelező**, egy laboróráról való hiányzást a hiányzás utáni első három héten be kell pótolni; előre is lehet a laborórákat pótolni
- a felmérőkön abban az esetben lehet részt venni, ha a diáknak nincs laborórákon hiányzása, ellenkező esetben csak a pótvizsgákon lehet résztvenni
- a vizsgajegyet pótvizsgákon is meg lehet szerezni, erre a két pótszesszióban van lehetőség
- a pótvizsgák is egy-egy feladatsorból fognak állni, amelyeket hasonlóan számítógépen kell leprogramozni Python programozási nyelvben
- a pótvizsgákon **nem lesz beszámítva** a pluszPont, a felmérő1, felmérő2
- az első pótvizsgaidőszak a félévi vakáció után van
- a második pótvizsgaidőszak az egyetemi év végén, ősszel van, amelyért az egyetem vizsgáztatási díjat számít fel

Könyvészet



Freud R., Gyarmati E., Számelmélet, Nemzeti Tankönyvkiadó, Budapest, 2000.



Cormen T.H., Leiserson C.E., Rivest R.L., Algoritmusok, Műszaki Könyvkiadó, Budapest, 2001.



Lovász L., Pelikán J., Vesztergombi K., Diszkrét matematika, Typotex, Budapest, 2006.



Rónyai L., Ivanyos G., Szabó R., Algoritmusok, Typotex, Budapest, 2004



Rosen K.H., Discrete Mathematics and its Applications, McGrawHill, New-York, 2012.



Magnus Lie Hetland M.L., Beginning Python: From Novice to Professional, 2nd edition, Apress, 2008.



<https://realpython.com/>



<http://nyelvek.inf.elte.hu/leirasok/Python/>

Bevezető

- diszkrét matematika → algoritmusok,
- minden algoritmust, programot Python programozási nyelvben írunk,
- ingyenesen letölthető verzió: **Python 3.10.7**,
<https://www.python.org/downloads/>
- a Python 1991-ben jelent meg, Guido van Rossum kezdte el fejleszteni,
- Python, tulajdonságok:
 - magasszintű adatszerkezetek: listák, ennesek (tuple), stringek, halmazok, stb.,
 - dinamikus típusrendszer: a típusinformációk kezelése futási időben történik
 - automatikus memóriakezelés,
 - objektum orientáltság,
 - rövid programok írása gyors és egyszerű
 - interpreter: a saját utasításait bemenő adatként kezeli, ezeket átalakítja a futtató gép utasításává, majd rögtön futtatja, ellentétben a fordító (kompilátor) típusú programozási nyelvekkel
 - a Python tulajdonképpen nem interpreter: byte kódot fordít és futtat

Python, elemi kódsorok

- a Python instalálása, majd indítása után megjelenik egy ablak egy kétsoros standard szöveggel (Python verzió szám, stb.) és a prompt: `>>>`
- a prompt után akifejezések, utasítások írhatóak, amelyeket a Python interpretere rögtön kiértékel:

Példák:

```
>>> print("Hello vilag!")  
Hello vilag!
```

```
>>> 251 + 965  
1216
```

```
>>> 2 ** 100  
1267650600228229401496703205376  
#A Python tetszolegesen nagy, egész számokkal is tud muveleteket  
#vegezni
```

```
>>> import math  
>>> math.sqrt(10)  
3.1622776601683795
```

Python, változók

A programozási nyelvek egyik alapfogalma a **változó** (variable):

- a változók értékek, adatok tárolását teszik lehetővé
- Python:
 - nem rendelkezik a hagyományos értelemben vett változókkal
 - objektum-referenciákon keresztül végzi az adattárolást
- az objektum-referencia jelentése programozási nyelvektől függően változik
- kezdő programozók számára a változó és referencia közötti különbségnek nincs nagy jelentősége,
- az egyszerűség kedvéért a továbbiakban a változó megnevezést fogjuk használni,
- a változó **értéke**: értékadás, értékmodosító utasításokkal határozzuk meg,
- a változó **típusa**: a típus alapján dől el, hogy a milyen fajta értékeket kezel/tárol,
- a változókkal végzett művelet sorok képezik a programírás alapjait,
- Python:
 - nincs explicit változódeklaráció,
 - a változódeklaráció **automatikus**, pl. értékadás során

Python lekérdezések

Példák:

```
>>> x = 251
>>> y = 965
>>> print(x + y)
1216
```

```
>>> x, y = 10, 4
>>> print(x / y, x // y , x % y)
2.5 2 2
```

```
>>> x, y = 17, 5
>>> print("valos osztas: ", x/y)
valos osztas: 3.4
>>> print("osztasi egeszresz: ", x // y)
osztasi egeszresz: 3
>>> print("osztasi maradek: ", x % y)
osztasi maradek: 2
```

Python alapfogalmak

- **operátorok:**
 - értékadó operátor: =
 - aritmetikai operátorok: +, -, *, /, //, %, ^, **
 - értékmodosító operátorok: +=, -=, *=, stb.
 - relációs operátorok: ==, !=, <, <=, >, >=
 - logikai operátorok: and, or, not
 - tagsági (membership) operátorok: in, not in
- **osztályok** (class) és **típusok** (type):
 - jelzik hogy milyen fajta adattal dolgozunk
 - kezdő programozók számára egy fogalmat jelölnek
 - egy adott érték típusa pont olyan fontos informatikai fogalom, mint az értéke
- **függvények** (function), és **metódusok** (methods):
 - egy függvény kiértékeli a zárójelben megadott kifejezést
 - beépített függvények, a programozó által írt függvények
 - pl. a **print** függvény kiértékeli a zárójelben megadott kifejezést, és az eredményt kiírja a standard kimenetre
 - pl. a **pow** függvénynek ha a két paramétere a és x akkor, meghatározza a^x értékét

Python alaptípusok/osztályok

Alaptípusok: int, float, str, bool

```
>>> type(103373189)
<class 'int'>
```

```
>>> type(106.909)
<class 'float'>
```

```
>>> type(True)
<class 'bool'>
```

```
>>> type('helo vilag')
<class 'str'>
```

```
>>> type("helo vilag")
<class 'str'>
```

Karakterláncok (str) jelölésére egyformán használhatjuk a ' , illetve " szimbólumokat

Típusok közötti átalakítások

Egész számmá alakítunk:

```
>>> int(23.11)
```

```
23
```

```
>>> int(23.0)
```

```
23
```

```
>>> int(-23.900)
```

```
-23
```

```
>>> int(10/3)
```

```
3
```

```
>>> int('23 szo')
```

```
Traceback (most recent call last):
```

```
  File "<pyshell#25>", line 1, in <module>
```

```
    int('23 szo')
```

```
ValueError: invalid literal for int() with base 10: '23 szo'
```

Típusok közötti átalakítások

Valós számmá alakítunk:

```
>>> float(15)
15.0
>>> float('23.67')
23.67
```

Karakterlánc típusúvá alakítunk:

```
>>> str(12)
'12'
>>> str(12.67)
'12.67'
>>> str(True)
'True'
```


A programírás lépései

- az interpreterbe beírt kifejezések, utasítások elvesznek, ha kilépünk a Pythonból,
- IDLE (integrated development environment): a Python standard **fejlesztői környezete**, programszerkesztésre, mentésre, állomány megnyitásra, futtatásra ad lehetőséget,
- a kódszerkesztőt a File/New File menüpontból lehet elindítani
- a kódszerkesztő menüpontjai:
 - File/New File - új állomány létrehozása;
 - File/Save - állomány mentése,
 - Run/Run Module vagy Ctrl/Fn + F5 - a **program futtatása** a Python shell-ből
 - stb.
- Python program: **script**
- gyakran használt fejlesztői környezetek:
 - Visual Studio Code
 - PyCharm
 - teszt