Tartalomjegyzék

- 1 Általános információk
- 2 Tananyag
- 3 1. el?adás (2010-02-08)
- 4 2. el?adás (2010-02-15)
- 5 3. el?adás (2010-02-22)
- 6 4. el?adás (2010-03-01)
- 7 5. el?adás (2010-03-08)
- 8 7. gyakorlat (2010-03-22)
- 9 8. el?adás (2010-03-29)
- 10 10. el?adás (2010-03-29)
- 11 11. gyakorlat
- 12 12. el?adás
- 13 13. el?adás
- 14 14. el?adás
- 15 Eredmények

Általános információk

- A tárgy el?adói és gyakorlatvezet?i: Lukács Ágnes, Kiss Tamás

  Email címek: {lagi, fadyga} KUKAC math PONT bme PONT hu

- Az el?adás id?pontja és helye: hétf? 08:15-9:00 H46.

  A gyakorlatok id?pontja és helye: hétf? 09:15-10:00 H57, péntek 9:15-10:00 H27

- **Követelmények**

Tananyag

A félév els? felében Python, illetve objektum-orientált programozás, második felében a C nyelv. Ajánlott irodalom Pythonhoz:

- **Python tutorial**
- http://www.greenteapress.com/thinkpython/thinkpython.html

A félév során a Python 2.6.4-es verzióját fogjuk használni, mely innen letölthet?:
http://python.org/download/

1. el?adás (2010-02-08)

Az el?adáson átismételtük a Sage-ben tanultakat, a következ? fogalmak kerültek el?:

- típus
- operátor
- vezérlési szerkezetek
- függvény definiálás

Az el?adáson átvett kódok a következ? linken elérhet?k: http://info.ilab.sztaki.hu/~kisstom/info2/

A következ? gyakorlatra két házi feladat van kit?zve:

- Írjunk Python-ban rekurzív függvényt, mely egy input egész n-re ellen?rzi a 3*n+1 algoritmus helyességét, és végül visszatér egy listával, mely tartalmazza azokat az egészeket, melyekre az algoritmus meghívódott. A helyesség ellen?rzése alatt azt értem, hogy a függvény addig hívja rekurzívan önmagát, amíg a kapott egész nem 1, ekkor egy üres listával return-nöl, egyébként meg a rekurzívan megkapott listával, amelyhez appendolja n jelenlegi értékét.
- Írjunk Python-ban függvényt, mely bemenetként egy n egészet vár, és az Erátoszthenészi szita algoritmusát használva kiírja n-ig a prímszámokat.

2. el?adás (2010-02-15)

Az el?adáson egy konvex burok meghatározó algoritmust vettünk. A lényege röviden: minden élre ellen?rizzük, hogy az él végpontjait nem számítva, az összes pont az él egy adott oldalán van. Azt, hogy az él adott oldalán van, az élb?l és a pontból kapott háromszög el?jeles területének el?jeléb?l tudjuk meg.

A gyakotlaton vettünk egy másik algoritmust, amely el?ször sorbarendezi a pontokat (x koordináta szerint), majd a fels? konvex burkot aszerint határozza meg, hogy az aktuális legszéls?höz képest "http://wiki.math.bme.hufeljebb"http://wiki.math.bme.hu van, akkor megtarjuk, ha lejjebb van, eldobjuk.

Ezen elgoritmusok forráskódjait megtaláljátok itt: http://math.bme.hu/~lagi/gyak2

A következ? gyakorlatra a következ? feladatok közül lehet választani:

- Jó ötlet alapján, az élek listájából a pontok listájának meghatározása (bonyodalom: 3 pont egy egyenesen, akkor mind a 3 él bent van.)
- (javasolt): olyan algoritmus, amely az x sorrend alapján elindul, majd a szög alapján "http://wiki.math.bme.hulegfels?"http://wiki.math.bme.hu pontot választja ki, az ilyenekb?l állítja össze a konvex burkot.
- Bármilyen más konvex burok algoritmus implementálása, pár sor magyarázattal együtt.

A programokat a következ? címre küldjétek: agicpp@gmail.com - JAVÍTÁS, ez a helyes cím!!

Határid?: 2010.02.28 23.59

3. el?adás (2010-02-22)

Az el?adás els? felében a python file és string osztályainak használatáról, fontosabb tagfüggvényeir?l volt szó, melyek segítségével egy file-ban eltárolt irányitott gráfot olvastunk be. A második felében átismételtük a mélységi bejárás algoritmusát, ami felhasználható arra, hogy meghatározzuk egy irányított körmentes gráf egy topologikus rendezését (a csúcsok olyan sorba rendezése, melynél minden él kisebb index?b?l vezet

nagyobb index?be). Ezekr?l b?vebben itt olvashattok:

http://fi.inf.elte.hu/adatszerkezet/ii_felev/graf/kricsi/

http://hu.wikipedia.org/wiki/Topologikus_sorrend

A házi feladat is ehhez kapcsolódik:

Készítsük el, egy irányított, körmentes gráf egy olyan szintekre bontását, melyekben az egy szinten lev? csúcsok között nem vezet él. A javasolt algoritmus egy iterációban meghatározza a gráf forrásait(azokat a csúcsokat melyekbe nem vezet irányított él), majd törli azokat. A feladathoz elkészítettem egy kódot(partition.py), amelyben már csak két függvényt kell megírni, bármilyen ett?l független jó megoldás beküldhet?. Az el?adás anyaga, a kód, valamint azok futtatása és kimenete itt megtalálható:

http://info.ilab.sztaki.hu/~kisstom/info2/3het

A program az el?adáson megadott formátumú file-t olvassa be, a futtatas_es_kimenet.txt-ben látható eredményt adja vissza.

A programokat a következ? címre küldjétek: fadyga@math.bme.hu

Határid?: 2010.03.07 23.59

4. el?adás (2010-03-01)

Az el?adáson az osztályokról volt szó. Az osztályok olyan struktúrák, amelyek tartalmazzák az adatokat és az ?ket kezel? függvényeket is (egységbezárás elve.) A python konstruktor __init__, reprezentáció __repr__, hossz __len__, nulla-e __nonzero__, + __add__, - __sub__, * __mul__, / __div__, tartalmazás __contains__ függvényeit vettük. Ezekr?l a függvényekr?l részletesen itt olvashattok a python manualban:

http://docs.python.org/reference/datamodel.html

Az el?adás végén az exception fogalma került el?, a gyakorlaton készítettünk saját Exceptiont is (a benti gépeken nem m?ködik a kód, de a Python IDLE-ben igen). Az exceptionok leírása itt található:

http://docs.python.org/tutorial/errors.html

A myset.py kód itt található:

http://www.math.bme.hu/~lagi/gyak4/myset.py

A modulon.py kód itt található:

http://www.math.bme.hu/~lagi/gyak4/modulon.py

HÁZI FELADAT: (határid?: 2010.03.18 20.00) Válasszatok egy tetsz?leges matematikai vagy adat tárolási objektumot. Ezeken értelmezzetek valamennyit a fenti függvények közül, (konstruktor, reprezentáció mindenképp), néhány operator overload-ot, ill. saját függvényeket, és készítsetek egy osztályt. A beadott kódban legyenek minta hívások is (tehát hogy hogyan lehet az osztályt használni.).

Például:

- kvaterniók (tárolás: együtthatók, m?veletek: összeadás, szorzás, kivonás, abszolút érték (hossz), konjugálás
- 2x2 -es mátrixok (tárolás: 4 db szám, m?veletek: összeadás, kivonás, szorzás, inverz, nulla-e)
- verem: push, pop: beteszek egy elemet, kiveszem a legfels?t (ha nem üres). Összeadás: a másodikat a fels? tetejére teszem. Megfordítás: fordított sorrendben legyenek benne az elemek. Hossz: hány db elem van benne, nonzero: üres-e, tartalmazás: benne van-e az elem.

Bármilyen egyéb, hasonló bonyolultságú dolog jöhet (és örülök neki)! A programokat a következ? címre küldjétek: agicpp@gmail.com

KONZULTÁCIÓ: A 6. héten elmarad a hétf?i el?adás. Azon a héten pénteken 2010.03.19-én lenne egy szabad termünk reggel 8-10-ig. Arra gondoltam, tartanék konzultációt python-ból (nagyjából milyen feladatok lennének, egy-két minta megoldás) + lehetne kérdezni. Mivel el?adás nincs azon a héten, gyakorlatot nem tartanék (mivel az csak az egyik csoportnak lenne), viszont ez lenne helyette.

UPDATE: A terem a T604. Mivel sokan mondtátok hogy jönnétek ezért akkor konzultáció lesz 8.30-tól.

A házi feladatokra kapott pontszámot itt nézhetitek meg:

http://spreadsheets.google.com/pub?key=tmL9WA0MdpOjN2VJ5d7MzJA&single=true&gid=1&output=html

A 2. és 4. házira egy-egy megoldás: http://www.math.bme.hu/~lagi/hazik/

5. el?adás (2010-03-08)

Az el?adáson osztályok öröklésével és a hozzá kapcsolódó fogalmakról volt szó. Bevezetésként készítettünk egy Point osztályt, mely kétdimenziós pontok reprezentálására alkalmas, illetve implementáltuk néhány egyszer? tagfüggvényét. Az óra második felében megvalósítottunk egy Sikidom osztályt, amely egy egységes felületet ad kés?bbi síkbeli osztályok létrehozásához (kör, háromszög). Az el?adáson és gyakorlaton vett kódok itt megtalálhatók:

http://info.ilab.sztaki.hu/~kisstom/info2/5het

HÁZI FELADAT:

Hozzunk létre örökléssel osztályokat, melyek személyek, és dolgozók adatit tudja tárolni, illetve néhány jellemz?jüket tudja számítani. Az ?sosztály neve legyen Szemely, __init__ tagfüggvénye kapja meg a személy életkorát illetve nevét. Az osztálynak legyen egy szabadnapok_száma tagfüggvénye, ami 20-szal térjen vissza, ha az illet? 35 évnél fiatalabb, egyébként 28-cal. Ebb?l származtassunk le egy Alkalmazott nev? osztályt, mely az életkor és név mellett még az illet? órabérét is eltárolja (az init függvényben használjuk a super függvényt az ?sosztály adatainak eléréséhez). Az alkalmazott osztályból származtassunk le két osztályt (félállású, illetve teljesállású) melyeknek legyen egy új fizetésszámít tagfüggvénye, félállású esetén 80 * oraber, teljesállású esetén 160*oraber legyen a visszatérési értéke. A teljesállású osztályból még származtassunk le egy fonok osztályt, melyhez egy premium nev? új adattagot vegyünk fel az el?z?ekhez, és a fizetésszámít függvényét módosítsuk a super függvény használatával úgy, hogy visszatérési értéke 160*oraber+premium legyen. Az el?z? linken találtok mintahívásokat az osztályokhoz.

Határid?: 2010.03.21 23.59

FIGYELEM: A 7. héten nem lesz el?adás, hanem mindkét alkalommal gyakorlatot tartunk, ezért kérném, hogy a hétf?i gyakorlat csoportja 8.30-ra jöjjön a H57-be, a pénteki pedig 8.30-ra a H27-be.

A korábbi házikra megoldásokat találtok az alábbi helyen:

http://info.ilab.sztaki.hu/~kisstom/info2/hazi_megoldasok

Az els? zárthelyi helye és id?pontja:

J102, április 1., csütörtök, 12:15-13:45

7. gyakorlat (2010-03-22)

A gyakorlaton vett feladatokat megtaláljátok itt: http://info.ilab.sztaki.hu/~kisstom/info2/7het

PÓT HÁZI: Horner-séma alkalmazásával számítsuk ki egy polinom helyettesítési értékét egy adott helyen. A polinomot együtthatóinak listájával azonosítsuk, ahol a lista k. eleme a k. fokú tag együtthatója. Például a

coef = [1.0,0.0,1.0,2.0,0.0,0.5] együtthatók és c = 3.0 helyettesítés esetén a következ? eredményt kapjuk: 185.5

8. el?adás (2010-03-29)

A kódokat megtaláljátok itt: http://math.bme.hu/~lagi/c1/

Az el?adáson áttekintettük a C nyelv alapvet? típusait, és foglalkoztunk a printf, scanf függvényekkel. A formátum stringeknél tudni kell a %d, %lf, %c formátumokat. A típusok közül tudni kell az int, short, long, char típusokat, mit jelent a signed-unsigned. A C nyelv operátorai: = == + - * / % ++ -- += -= *= /= | & ^ && || ?: (konverzió)

Házi feladat: Kérjük be egy tetraéder 4 csúcsát (12 szám). Számítsuk ki és írjuk ki a térfogatát (pl vegyesszorzat/6) Határid?: 2010.04.11. 20.00

10. el?adás (2010-03-29)

A kódokat megtaláljátok itt: http://math.bme.hu/~lagi/c2/

Az el?adáson megismerkedtünk a C nyelv vezérlési szerkezeteivel: if, while, for, do-while, switch, majd foglalkoztunk a pointer típussal. A gyakorlaton néztünk egy példát a switch-re, majd kiírtuk egy szám osztóit és a prímfelbontását.

Házi feladat: Kérjük be két számot, írjuk ki a legnagyobb közös osztójukat. Határid?: 2010.04.25. 20.00

11. gyakorlat

Gyakorlaton megírtunk egy getline nev? függvényt, ami az stdin-r?l beolvas a paraméterként megkapott tömbbe num - 1 karaktert, és az egészet lezárja a '\0' karakterrel (a num integert szintén paraméterként kapja meg).

Házi feladat: legyen adott egy szárazföldb?l és vízb?l álló térkép, melyet egy karakterekb?l álló 2 dimenziós tömbként tárolunk el, ahol a '.' karakter vizet, a 'X' szárazföldet jelent. A vízszintesen vagy függ?legesen szomszédos szárazföldek együtt szigetet alkotnak. A térkép legyen globális változó, mérete egy adott konstans (pl 1000*1000). Feladat, hogy megszámoljuk a térképen lev? szigetek számát. Ehhez egy sullyeszt függvényt kell megírni, amelyet a következ? módon deklarálunk:

void sullyeszt (int x, int y);

ahol x,y a térkép egy koordinátája. A függvény a következ? módon m?ködjön: ha nem a térképen lev? koordinátát kap lépjen ki, ha nem szárazföldet talál lépjen ki, egyébként az adott területet süllyessze el (a tömb i,j eleme legyen '.'), és hívja meg magát rekurzívan a terület vízszintes és függ?leges szomszédaira. A main függvény olvassa be a sorok számát, majd a szigetet, és a sullyeszt függvény használatával számolja össze a szigeteket. Beolvasáshoz használhatjuk az általunk megírt getline függvényt, de egyéb módon is (pl fgets) lehetséges. Az alábbi linken találtok egy megoldás kezdeményt, illetve példát a kimenetre:

http://info.ilab.sztaki.hu/~kisstom/info2/11het

Határid?: május 9, 24:00

12. el?adás

Szó volt a malloc függvény használatáról, a statikus és dinamikus tömbökr?l. Elhangzott, hogy "http://wiki.math.bme.hutömb neve = a 0. elem címe"http://wiki.math.bme.hu és ennek következményeit tekintettük át (pointer aritmetika). Megnéztük, hogyan lehet C-ben struktúrát definiálni. Az el?adás 2. részében a file-kezelés legfontosabb függvényeit néztük meg: fopen, fclose, fprintf, fscanf, fread, fwrite. A gyakorlaton vett programokat feltettem ide: http://www.math.bme.hu/~lagi/c3/

HÁZI FELADAT: Az órán vett fájl író program mintájára, kell írni egy olyan programot, amely egy valamilyen dolog adatait kiírja egy szöveges fájlba, és egy másikat, ami visszaolvassa az adatokat. Az, hogy mir?l legyen az adat, mindenki találjon ki magának egy témát, és nem szeretnék egyforma megoldásokat látni a házikban! Példa:

- autó - típus, évjárat, motor térfogat, szín
- ingatlan - címe, telek terület, hány nm, ár
- CD - címe, el?adó, kiadás éve, hány számot tartalmaz, stílus

Stb (a fentiek már nem választhatók, csak példák). Mivel úgy azért túl könny? lenne, ha a teljes visszaolvasó programot kitenném, ezért ezt nem teszem, csak a legfontosabb sort írom ide, segítségként:

```
while(fscanf(f,"http://wiki.math.bme.hu%30s\n%2d%20s\n"http://wiki.math.bme.hu,nev,&kor,szak)!=
```

Ez a rész addig olvassa a fájlt, amíg nem ér a végére (-1=EOF end of file kód). Opcionálisan a bináris fájl programot is lehet mintául venni, ha az valakinek jobban tetszik. A fájl írását megtaláljátok a példák között, a visszaolvasás f? sorát itt is megadom:

```
while(fread(&szem,sizeof(szem),1,f)!=0)
```

Az fread azt adja vissza, hogy mennyi adatot sikerült kiolvasnia sikeresen. Ha 0, akkor vége van a fájlnak.

Tehát a beadandó:

- A fájl írását végz? program az adott, önállóan kiválasztott témára. Mindenki legalább 5 féle adatot írjon ki, ezek között legyen string, szám is. (Ha lebeg?pontos számot használsz, akkor pl %8.3lf

-ként tudod megadni neki).
- A fájl visszaolvasását végz? program.
- Egy minta adat fájl, legalább 5 értelmes adattal (ezzel fogom kipróbálni a fájl visszaolvasó programokat.)

Javaslom, hogy mindenki nézze át a bináris fájlt is, mert ZH-ra azt is tudni kell! A ZH a 14. héten lesz, Tamással megbeszéltük. A pontos id?pontot még nem tudjuk.

13. el?adás

Az el?adáson megismertük a láncolt lista illetve a bináris keres?fa adatszerkezeteket, készítettünk implementációt listaelemre, faelemre illetve az adatszerkezetek által támogatott m?veletekre (lista esetén keresés, beszúrás, törlés; keres?fa esetén inorder bejárás, minimális elem kiválasztása). Az el?adáson vett kódok itt megtalálhatóak:

http://info.ilab.sztaki.hu/~kisstom/info2/13het

A 2. ZÁRTHELYI helye KA61, id?pont május 13, 12:15-13:45.

HÁZI FELADAT: Készítsünk egy rendezett láncolt lista adatszerkezetet, amely int típusú adatot tud tárolni, illetve annak beszur m?veletét, amely létrehoz egy a paraméterként megkapott kulcsú listaelemet, és rendezést megtartva beszúrja a listába. A megoldáshoz készítsünk egy main-t, amelyben teszteljük a függvényt, és írassuk is ki a lista elemeit. HATÁRID?: május 14, 23:59

14. el?adás

Az el?adáson megnéztük, milyen lehet?ségei vannak a C++-nak a C nyelv képességein túl. Ez a rész már nem a zh anyaga.

A gyakorlaton egy kapcsolós feladatot néztünk, 2 példa megoldás:

http://www.math.bme.hu/~lagi/snapper

A PÓT ZH május 19.-én, 12:15-13:45 között lesz, a Z105-ben.

Eredmények

A félévi eredmények összesítését itt nézhetitek meg:

http://spreadsheets.google.com/pub?key=tmL9WA0MdpOjN2VJ5d7MzJA&single=true&gid=1&output=html

Ha valakinek pótolnia kell valamit (például nem adott be elég házifeladatot), azt még a héten péntekig tegye meg.