



ALGORÍSMICA

Presentació de l'assignatura

Contactes

+ Professors de Teoria: Jordi Vitrià, Mireia Ribera
Email: jordi.vitria@ub.edu, ribera@ub.edu

+ Professors de Pràctiques: Pablo Laíz, Carolina Bonnín, Ramon Navarro
Email: laizpablo@ub.edu, carolinabonnin@ub.edu, ramon.nb@gmail.com

+ Auxiliar docent: Pere Gilabert (sessions específiques per a aquells alumnes que vagin amb més retard o que no hagin programat mai i tinguin dificultats a l'inici).

Què és aquesta assignatura?

Aquesta assignatura està dirigida a donar la formació bàsica als estudiants sobre **l'anàlisi i disseny d'algorismes, tant des d'un punt de vista teòric com aplicat.** No s'assumeix cap formació prèvia en programació de l'estudiant.

Què s'espera dels estudiants matriculats?

Els estudiants han de participar de forma activa durant les classes magistrals de **teoria** (1,5 hores a la setmana).

Durant les hores teòrico-pràctiques (o de **problemes**, 2 hores cada dues setmanes) hauran de dissenyar solucions algorísmiques als problemes plantejats pels professors.

Durant les hores presencials de **pràctiques** (2 hores cada dues setmanes) hauran de programar de forma individual una sèrie d'exercicis pràctics.

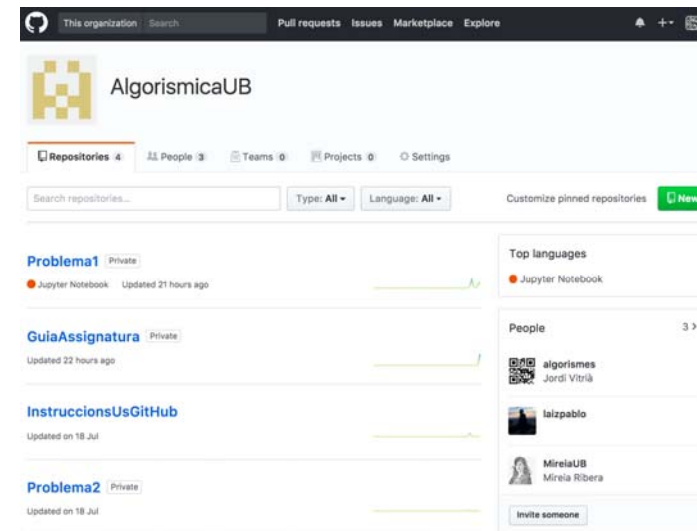
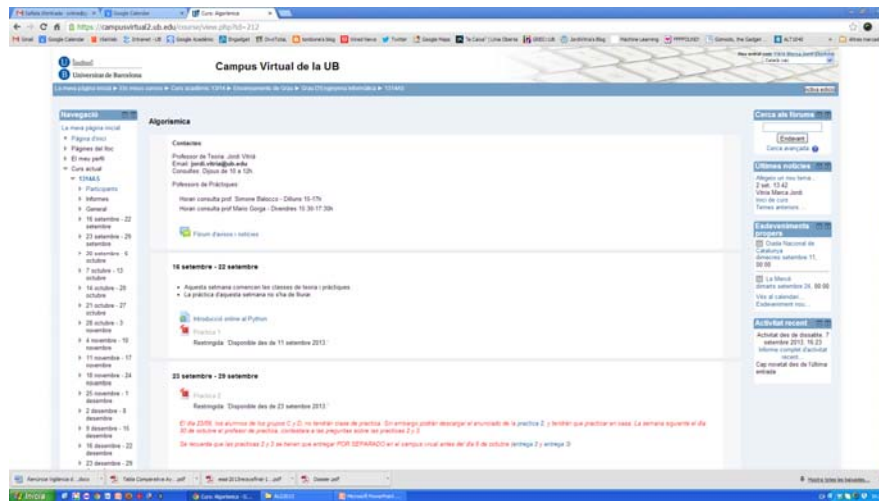
Les hores no presencials de l'assignatura (4 hores a la setmana) les han de dedicar a l'estudi i a la preparació dels problemes i pràctiques.

Programarem?

Tot i que en aquesta assignatura no és estrictament necessari programar, ho farem amb un llenguatge d'alt nivell: **Python**.

Com s'organitza l'assignatura?

Usarem dues eines per distribuir la informació i organitzar la feina: el campus virtual de la UB i GitHub.



El campus virtual

- Apunts de teoria
- Enunciat de les pràctiques
- Lliurament de les pràctiques
- Avaluacions de les proves i pràctiques

Poseu-hi una fotografia digital vostra
<http://campusvirtual2.ub.edu>

GitHub

- Enunciats dels problemes
- Portfoli individual dels problemes resolts

El punt de partida d'un estudiant és crear un compte personal a GitHub amb el format *nomCognom* (per exemple, *mireiaRibera*)

La creació d'aquest compte és gratuïta i es pot fer a **<https://github.com>**.

Un compte personal gratuït permet crear un nombre il·limitat de repositoris públics. Els repositoris públics són visibles per al públic en general. Normalment s'utilitzen per desenvolupar programari lliure amb qualsevol tipus de llicència. Un repositori és, doncs, un lloc on es poden emmagatzemar fitxers d'un projecte utilitzant el control de versions Git.

Com s'avaluarà l'assignatura? (I)

L'assignatura seguirà un esquema **d'avaluació continuada**, amb dos elements principals: proves presencials i lliurament remot d'exercicis.

- **Lliurament via web de pràctiques (LP):** Els professors, amb periodicitat quinzenal, proposaran una sèrie de pràctiques que hauran de ser lliurades via web per part de l'alumne dins el període assenyalat pel professor. Cada un dels lliuraments serà avaluat pel professor amb una nota que pot anar de 0 (nota mínima) a 10 (nota màxima). En cas de no lliurar les pràctiques **dins el període assenyalat**, l'alumne obtindrà un 0. La nota final (LP) de la part de lliurament de pràctiques serà la mitja de tots els lliuraments.
- **Proves presencials (PP):** durant el curs, l'alumne realitzarà varies proves escrites (teòrico-pràctiques) davant del professor. Les proves s'avaluaran amb una nota de 0 (nota mínima) a 10 (nota màxima). La nota final (PP) d'aquesta part serà la mitja de totes les proves realitzades (una prova no realitzada = 0).

La nota segons l'avaluació continuada (NF) es calcularà de la següent manera:

- Si (LP > 4,0 i PP > 4,0): $NF = PP * 0,6 + LP * 0,4$
- Sinó, $NF = \min(4,0, PP * 0,6 + LP * 0,4)$.

Com s'avaluarà l'assignatura? (II)

- Durant la segona prova presencial (Gener) es donarà l'opció de presentar-se de tota l'assignatura o només de la segona part.
- Tots aquells alumnes que obtinguin una $NF \geq 3,5$ tenen dret a **una reavaluació** al cap d'un dies de la publicació de NF. La reavaluació serà equivalent a un examen final. En aquests casos, la nota final de l'assignatura serà la nota de la reavaluació.

I el lliurament de problemes...?

No hi ha una activitat pròpia de lliurament de problemes, però l'alumne anirà construint un portafolis de problemes que el professor pot revisar en qualsevol moment.

La part pràctica de les **proves presencials** estarà basada en **aquests problemes**.

GitHub proporciona informació sobre QUAN s'ha resolt els problemes. Per valorar aquesta activitat el professor tindrà en compte tant el QUÈ com el QUAN.

Calendari de proves

El 6 de novembre de 15h a 18h es farà la primera prova presencial. És un període no lectiu.

La segona prova i la reavaluació es faran durant el mes de gener.

Apunts als exàmens?

U

Universitat de Barcelona

La meua pàgina inicial

► Els meus cursos

► 15/16

► Ensenyament de Graú

► Graú De Matemàtiques

► 15/16AS

Configuració

Administració del curs

Activa edició

Edita paràmetres

Usuaris

Filtres

Informes

Qualificacions

Competències

Insignies

Còpia de seguretat

Restaura

Importa

Rebreix

Banc de preguntes

Canvia rol a...

Configuració del meu perfil

Algorismica

Professor de Teoria:

Jordi Vitrà (jordi.vitra@ub.edu). Horari consulta : Dilluns 12-14h

Professors de Pràctiques:

Mireia Ribera (mireia@ub.edu). Horari consulta : Dimarts de 12-13h

Carles Franquesa (carlesfranquesa@ub.edu)

Simone Balocco (simone.balocco@ub.edu)

Recursos de l'assignatura

Xuleta Python

Xuleta Python

Instruccions Python

Fòrum d'avisos i notícies

Fòrum de pràctiques

Filtres de pràctiques de laboratori

14 setembre - 20 setembre

Aquesta setmana, tot i que el que fareu no és avaluable i no conta a la nota, heu d'assistir a les classes

ALG0

ALG1

Notebook 1: Python i Python

Aquest és l'únic material que podreu portar als exàmens de pràctiques. Si hi trobeu a faltar alguna cosa, es pot considerar la seva incorporació.

Introducció al Python

Python

www.python.org

Python és un llenguatge de programació genèric de gran difusió que pot ser usat en qualsevol tipus de tasca que no requereixi accés directe al hardware del sistema o processament en temps real o que impliqui el desenvolupament i manteniment d'una gran infraestructura software per part de molta gent. La principal causa d'aquestes limitacions (i també dels seus avantatges en el altres àmbits) és que té un sistema dèbil de control de la semàntica estàtica del codi.

És un llenguatge simple, amb una corba d'aprenentatge ràpida, i disposa d'un gran nombre de mòduls que fan possible la seva aplicació a molt diverses àrees d'aplicació, des del càlcul científic al desenvolupament de plataformes web.

Python és un llenguatge interpretat. Això vol dir que les instruccions que el programador escriu no són instruccions que la plataforma que el conté pugui entendre directament i per tant necessiten ser traduïdes per un altre programa a instruccions comprensibles per la màquina. Aquesta característica el fa flexible.

Elements Bàsics

Programa Un programa en Python és una seqüència de definicions i comandes.

Comanda Una comanda és una instrucció directa a l'interpreter per fer alguna cosa. El procés per determinar el resultat d'una comanda es diu **avaluació**.

Literal Un literal és una entitat el valor de la qual és ella mateixa i que per tant no cal avaluar. 3 és un literal.

Objectes Els objectes són les entitats bàsiques que manipula Python. Tots els objectes tenen un tipus que defineix quines operacions podem aplicar-les. Els tipus poden ser **escalars** o **no escalars**. Els primers són indivisibles, els segons tenen estructura interna.

Tipus escalars Python té varis tipus escalars:

```
int S'usa per representar els nombres enters: 3, 1245 o -23.
int S'usa per representar els nombres enters de longitud arbitrària: 334676346634547L.
float S'usa per representar els nombres reals: 3.0, 1245.2325 o -23.56.
bool S'usa per representar valors els valors Booleanns True o False.
None És un tipus amb un únic valor.
```

Operadors Els operadors ens permeten combinar objectes, formant expressions, cada una de les quals denota un objecte, anomenat valor, d'un determinat tipus. L'expressió `3 + 2` denota l'objecte 5 de tipus int. Els operadors dels tipus int i float són:

```
i+j S'usa per representar la suma. Si tant i com j són int, el resultat és int. Si algun dels dos és float, el resultat és float.
i-j S'usa per representar la resta. Si tant i com j són int, el resultat és int. Si algun dels dos és float, el resultat és float.
i*j S'usa per representar el producte. Si tant i com j són int, el resultat és int. Si algun dels dos és float, el resultat és float.
i/j S'usa per representar la divisió entera (retorna el quocient i s'obliga del reste): el valor de 6 / 2 és 3 i el valor de 6 / 4 és 1.
```

```
i/j S'usa per representar la divisió. Si tant i com j són int, el resultat és int. Si algun dels dos és float, el resultat és float.
```

```
i%j S'usa per representar la resta (o mòdul) de la divisió entera de i per j.
```

```
i**j S'usa per representar  $i^j$ . Si tant i com j són int, el resultat és int. Si algun dels dos és float, el resultat és float.
```

```
>, <, >=, <=, ==, != Són els operadors de comparació, el resultat dels quals és de tipus bool.
```

Tots aquests operadors segueixen l'ordre de precedència habitual.

Els operadors del tipus bool són:

```
a and b Si tant a com b són True, el resultat és True. Si algun dels dos és False, el resultat és False.
```

```
a or b Si a o b són True, el resultat és True. Si els dos són False, el resultat és False.
```

```
not a Si a és True, el resultat és False. Si a és False, el resultat és True.
```

Variables Les variables, en Python, són un mecanisme per associar un nom a un objecte. Considerem aquestes comandes:

```
1 pi = 3.14
2 radi = 11.2
3 area = pi * (radi ** 2)
```

El que fa Python és, primer, associar els noms pi i radi a dos objectes de tipus float i després associar el nom area a un altre objecte de tipus float. Si a continuació escrivim:

```
1 radi = 0.0
```

Python deslliga l'associació anterior del nom radi i l'associa a un objecte diferent de tipus float. En Python una variable és només un nom. Un objecte pot tenir un nom, diversos noms, o cap nom associat amb ell.

Els noms de les variables poden estar formats per lletres majúscules i minúscules, dígitos (tot i que no poden estar el principi), i el caràcter `_`. Les variables Jordi i Jordi són noms diferents. Hi ha un conjunt de paraules que no es poden usar com a noms de variables perquè estan reservades: `and, as, assert, break, class, continue, def, del, elif, else, except, exec, finally, for, from, global, if, import, in, is, lambda, not, or, pass, print, raise, return, try, with, while i yield`.

Comentaris Podem posar comentaris sobre el codi que escrivim usant el símbol `#` a principi de línia:

```
1 # Càlcul de l'àrea d'un cercle
2 pi = 3.14
3 radi = 11.2
4 area = pi * (radi ** 2)
```

Assignació múltiple Python ens permet assignar noms als objectes en *paral·lel*:

```
1 x, y = 2, 3
2 x, y = y, x
```

Si executem aquestes instruccions, el contingut final a x és 3 i a y és 2.

El tipus Str

El tipus no escalar **str** serveix per representar seqüències de caràcters o **strings**. Els literals de tipus **str** es poden escriure de dues maneres: `'abcd'` o `"abcd"`. El literal `'\t'` representa el caràcter i no el nombre! La longitud d'un **string** es pot saber amb la funció `len('abc')` és 3. Hi ha alguns operadors numèrics que es poden aplicar també a aquest tipus. Aquesta abstracció s'anomena **sobrecàrrega**. Considerem:

```
1 b,c = 'a' + 'a', 'a'*3
```

Llavors, el contingut de b és 'aa' i el de c és 'aaa'.

Indexació La indexació és l'operació que ens permet extreure caràcters individuals d'un **string**. Per exemple, `'abc'[0]` és 'a' i `'abc'[2]` és 'c'. És important observar que donat un **string** s a la primera posició és 0 i l'última `len(s)-1`. Si escrivim `'abc'[3]` es produirà un error per voler accedir a una posició inexistent. Els índexos negatius s'interpreten en ordre invers: `'abc'[-1]` és 'c'.

Slicing L'**slicing** és l'operació que ens permet extreure **substrings** de qualsevol mida. Si s és un **string**, `s[start:end]` denota el substring que comença a la posició start i acaba a end-1: `'abc'[1:3]` és 'ab'.

Input Python té dues funcions per obtenir dades del teclat de l'ordinador: `input` i `raw_input`. Les dues fan que el programa s'aturi fins que l'usuari introdueixi un **string** pel teclat de l'ordinador. `raw_input` tracta l'entrada com un **string** i `input` considera que el que ha entrat l'usuari és una expressió Python, que avalua per inferir-ne els tipus i extreure'n el valor.

Condicionals

Fins ara hem vist seqüències lineals d'instruccions, que són executades per l'interpreter Python una darrera l'altra. Si volem un esquema d'execució condicional, en arbre, necessitem especificar tres parts:

1. Un test (que és una expressió de tipus bool).
2. Un bloc de codi que s'executi quan l'expressió prengui el valor True.
3. Un bloc de codi, opcional, que s'executi quan l'expressió prengui el valor False.

Les instruccions que implementen aquest esquema s'anomenen **condicionals** i tenen aquesta forma:

```
1 if a > 3.0:
2     b = 0.0
3 else:
4     b = True
```

Python usa l'estructura visual del codi (definida per les indentacions de cada línia) com a part de la seva semàntica. Concretament, el codi que s'executa quan l'expressió booleana pren un determinat valor ha d'estar en un nivell superior d'indentació que el test. L'indentació de Python són 4 caràcters en blanc.

Aquesta regla es pot aplicar a múltiples nivells:

```
1 if x%2 == 0:
2     if x%3 == 0:
3         print 'Divisible per 2 i per 3'
4     else:
5         print 'Divisible per 2 i no per 3'
6 elif x%3 == 0:
7     print 'Divisible per 3 pero no per 2'
8 else:
9     print 'No es divisible ni per 2 ni per 3'
```

13

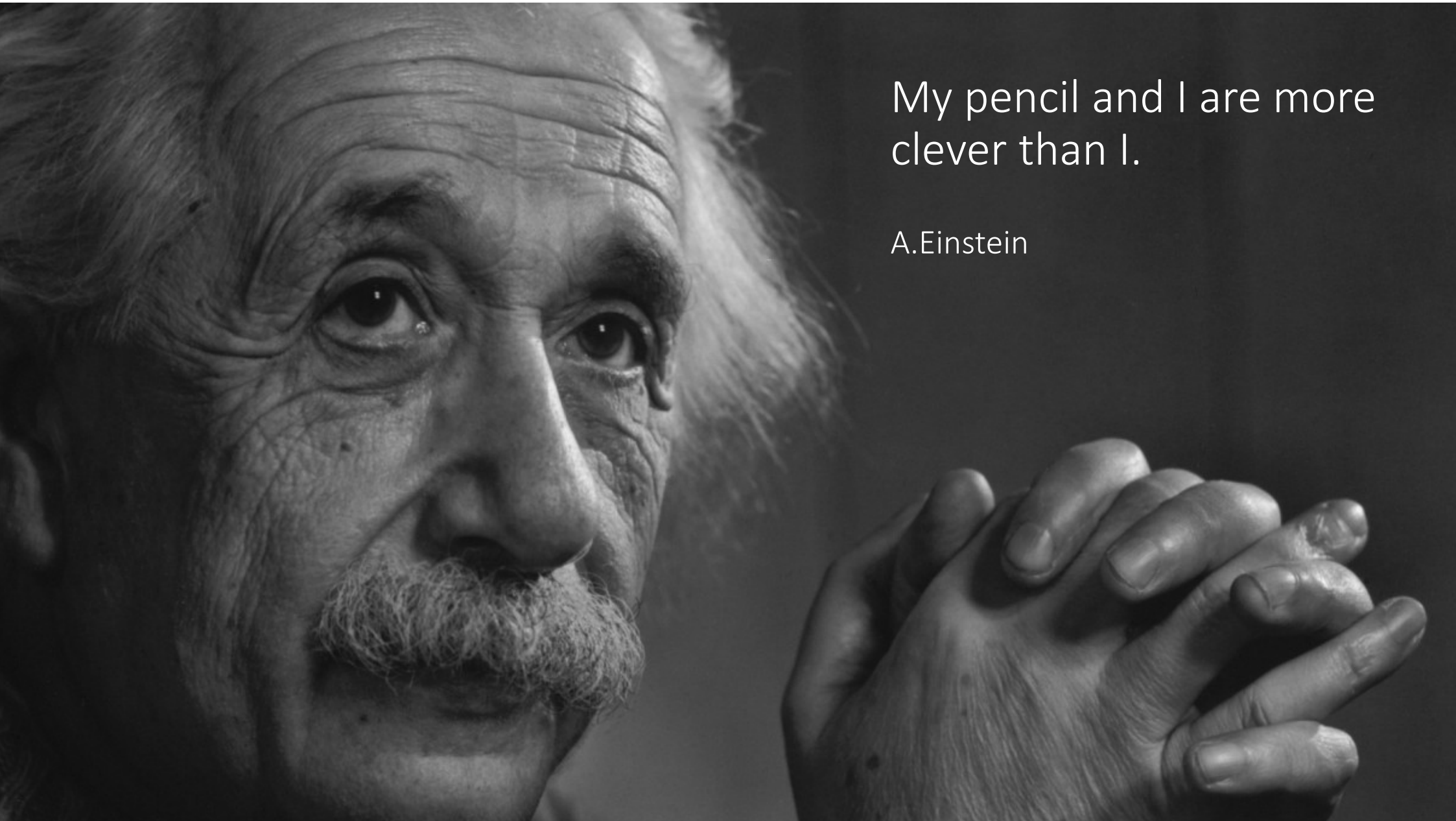
Bibliografia

Algorísmica

1. T. H. Cormen [et al.] Introduction to algorithms, MIT Press, 2001.
2. S. Dasgupta. Algorithms, McGrawHill, 2006.
www.cs.berkeley.edu/~vazirani/algorithms/all.pdf
3. V. Levitin, Introduction to the Design and Analysis of Algorithms, ISBN: 0-201-74395-7, Addison-Wesley (2nd edition)
4. S. Skiena. The Algorithm Design Manual, Springer; 2nd edition (August 21, 2008), Language: English, ISBN-10: 1848000693.

Python

1. A. Downey, J. Elkner and C. Meyers. How to Think Like a Computer Scientist. Learning with Python.
<http://greenteapress.com/thinkpython/thinkCSpy/>



My pencil and I are more
clever than I.

A.Einstein

$$\psi(g) = g \quad g = \text{Dlog } f_u \quad ?$$

$$\Leftrightarrow \exists \alpha \in \mathbb{C} \quad M = X(E|F_\infty)$$

$$X^2 + aX + b \quad X^2 + X + 1 \quad \text{for some } u \quad \text{SE 211 @ cam}$$

$$+ \frac{a}{1+a} \quad \phi_1 \quad (\mu_{f,\infty}) \quad H^0 \quad U_\alpha \quad \mathbb{E} = \{ (x^0, x^1) \mid x^0 \in \mathbb{Q} \}$$

$$\frac{z^2 + 2z}{z+1} = \alpha \frac{(z+1)}{(z+1)} \quad f(T) \quad F \quad (xy)^m = \frac{1}{m} (z^{\frac{m+1}{2}} + y^{\frac{m+1}{2}})$$

$$\Delta(1) \quad \frac{1}{T} \quad N f_u = f_u \quad \sum_{n \in \mathbb{Z}} a_n T^n = \sum_{n \in \mathbb{Z}} p_n T^n$$

$$\phi_r \quad D = (1+T) \frac{d}{dT} \quad N(\text{turns}) \quad \frac{1}{2} \alpha^2 + 2\alpha$$

$$0 \rightarrow \text{sl}(E/K) \rightarrow H^1(\mathbb{P}^1(\mathbb{C}), E_{\text{reg}}) \rightarrow \dots \quad \alpha \quad \beta$$

$$\psi(f) \quad (1+i) \quad H^1(\mathbb{C}) \rightarrow \pi_1 \quad A_\alpha \sum \alpha_i \pi_i$$

$$N(\mathbb{P}^1) = 2 \quad 3 \quad \text{sl}(E/K) \rightarrow \text{sl}(E/K) \quad A \psi$$

$$\text{Ker}(f) = H^1(\Delta, E_{\text{reg}}(\mathbb{Z}^n)) \quad 1 = \dots$$

$$\int \psi dN(\alpha, \beta) = \int \psi dN(\alpha, \beta) \quad X^2 + X + 1 = 0 \quad S_3 = S \rightarrow S_2 = S$$

$$X^2 + X + 1 = 0 \quad S_3 = S \rightarrow S_2 = S \quad S_3 = \emptyset \rightarrow \emptyset^2$$

$$X^2 + X + 1 = 0 \quad x = x+1 \quad \frac{1}{2} \alpha^2 + 2\alpha$$

$$X^2 + X + 1 = 0 \quad T = \sum a_n T^n \quad \sigma_2$$

$$X^2 + X + 1 = 0 \quad \sigma_2 \quad \sigma_8$$

$$X^2 + X + 1 = 0 \quad \sigma_2 \quad \sigma_8 \quad \sigma_{16}$$

$$X^2 + X + 1 = 0 \quad \sigma_2 \quad \sigma_8 \quad \sigma_{16} \quad \sigma_{30}$$

$$X^2 + X + 1 = 0 \quad \sigma_2 \quad \sigma_8 \quad \sigma_{16} \quad \sigma_{30} \quad \sigma_{40}$$

$$X^2 + X + 1 = 0 \quad \sigma_2 \quad \sigma_8 \quad \sigma_{16} \quad \sigma_{30} \quad \sigma_{40} \quad \sigma_{50}$$

$$X^2 + X + 1 = 0 \quad \sigma_2 \quad \sigma_8 \quad \sigma_{16} \quad \sigma_{30} \quad \sigma_{40} \quad \sigma_{50} \quad \sigma_{60}$$

