

Algoritmos

Computación 2021

FaMAF

17 mar. 2021

Contenidos

- 1 *Generalidades de la materia*
 - Idea general del curso
 - Condiciones administrativas

- 2 *Programación*
 - Lenguajes de programación
 - Paradigmas de programación
 - Algoritmos

- 3 *Python*
 - Lenguajes de programación
 - Python para iniciados

Generalidades de la materia

De qué se trata "Computación"?

El rol de las tecnologías en la educación moderna

Al contrario de lo que suele pensarse, los nativos digitales no entienden más sobre cómo funcionan las tecnologías no son mejores en multitarea.

- The digital native is a myth
- Los jóvenes hacen un uso superficial de las nuevas tecnologías

Preparando a las nuevas generaciones

Los trabajos del futuro (proyección para 2030 en UK):

- La automatización lleva a requerir nuevos "skills" (aptitudes)
- Proyección para 2030:
- 91% + en IT y programación
- 69% + en herramientas digitales básicas
- 40% + en creatividad
- 33% + en emprendedurismo e iniciativa
- 33% + en liderazgo y manejo de equipos de trabajo

La vida social del futuro:

- seguridad en internet
- ciudadanía digital

Recursos computacionales

Entonces,

- qué se puede hacer?
- que no se puede hacer?

Recursos computacionales

Entender las nuevas tecnologías de cómputo ayuda a:

- realizar tareas repetitivas de manera rápida
- realizar tareas tediosas con poco esfuerzo
- resolver problemas que de otra forma no se pueden resolver

Entender las nuevas tecnologías de la información ayuda a:

- usar y evaluar fuentes de información en internet efectivamente
- capacidad crítica

Recursos computacionales

Qué problemas podemos resolver?

- Ajedrez, Go
- Ordenar
- Encontrar
- Simular
- Recomendar (ML)
- Clasificar (ML)

Regularidad y promoción

Para regularizar:

- Dos parciales y un recuperatorio

Para promocionar:

- promedio > 7
- parciales > 6
- tener aprobadas las correlativas

Forma de trabajo

- aula virtual
- guías (no vamos a pedir)
- trabajo en casa
- computadoras y videoconferencias
- python
- linux
- no vamos a enseñar a "usar una PC", sino a resolver problemas que requieren usar una PC. El uso de la PC (crear y mover archivos y carpetas, buscar, organizar, etc.) se aprenderá como un subproducto.

Programación

Computación...

Conocimiento puede ser:

- Declarativa
- Imperativa

La implementación con un lenguaje puede ser ("paradigmas de programación"):

- Imperativa (estructurada, procedimental, modular)
- Orientada a objetos
- Declarativa
- Reactiva

¿Qué es un algoritmo?

Encuentre su globo



Las torres de Hanoi



- Mover un disco a la vez
- Solo se puede mover el disco superior de una pila
- No se puede poner un disco arriba de otro más pequeño

Algoritmos

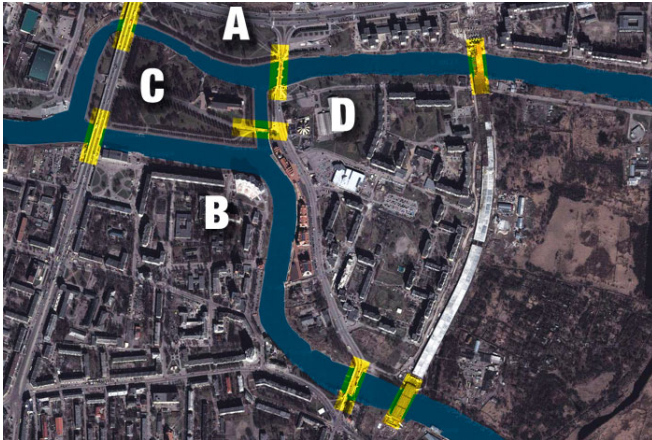
Definition

Un algoritmo es un conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

Los puentes de Königsberg



Los puentes de Königsberg



Algoritmo de Euclides

El algoritmo de Euclides permite encontrar el máximo comun divisor entre dos números naturales a y b , y se denota $mcd(a, b)$.

Consideremos los siguientes pasos:

- Dados a y b , tomar $r_0 = b$
- Defino r_1 como es resto de dividir a por r_0
- Si $r_1 = 0$, entonces r_0 es el $mcd(a, b)$
- Si no, defino r_2 como es resto de dividir r_0 por r_1
- Si $r_2 = 0$, entonces r_1 es el $mcd(a, b)$
- ...
- Defino r_{i+1} como es resto de dividir r_{i-1} por r_i
- Si $r_{i+1} = 0$, entonces r_i es el $mcd(a, b)$
- ...

Algoritmo de Euclides

El algoritmo de Euclides se puede simplificar un poco:

- 1 Dados a y b , definir $k = 1$, $r_0 = b$, $r_1 = a \bmod r_0$
- 2 Si $r_k = 0$, detener y devolver como solución $\text{mcd}(a, b) = r_{k-1}$
- 3 $r_{k+1} = r_{k-1} \bmod r_k$
- 4 $k \leftarrow k+1$
- 5 ir al paso (1)

Algoritmo de Euclides

- ➊ Dados a y b , definir $k = 1$, $r_0 = b$, $r_1 = a \bmod r_0$
 - ➋ Si $r_k = 0$, detener y devolver como solución $\text{mcd}(a, b) = r_{k-1}$
 - ➌ $r_{k+1} = r_{k-1} \bmod r_k$
 - ➍ $k \leftarrow k+1$
 - ➎ ir al paso (1)
-
- ➊ Dados a y b , definir $k = 1$, $r_0 = b$, $r_1 = a \bmod r_0$
 - ➋ Repetir:
 - ➌ $r_{k+1} = r_{k-1} \bmod r_k$
 - ➍ Si $r_k = 0$, devolver r_{k-1} y terminar la repetición.
 - ➎ $k = k + 1$

Algoritmo de Euclides

- ➊ Dados a y b , definir $k = 1$, $r_0 = b$, $r_1 = a \bmod r_0$
- ➋ Repetir:
- ➌ $r_{k+1} = r_{k-1} \bmod r_k$
- ➍ Si $r_k = 0$, devolver r_{k-1} y terminar la repetición.
- ➎ $k = k + 1$

Algoritmo de Euclides

```
r_k_menos_1 = a
r_k = b

while r_k != 0:
    r_k_mas_1 = r_k_menos_1 % r_k
    r_k_menos_1 = r_k
    r_k = r_k_mas_1
```


Algoritmo de Euclides

```
def MCD(a, b):  
    while(b != 0):  
        temp = b  
        b = a % b  
        a = temp  
    return a
```

Algoritmo de las torres de Hanoi

```
torre(disco, origen, inter, destino)
```

```
IF disco es igual a 1, THEN
```

```
    mover el disco del origen al destino
```

```
ELSE
```

```
    torre(disco - 1, origen, destino, intermedio)
```

```
    mover el disco del origen al destino
```

```
    tower(disco - 1, intermedio, origen, destino)
```

```
END IF
```

```
END
```

Algoritmo de Haze

Supongamos que para un entero positivo cualquiera, a_0 , definimos:

$$a_{k+1} = \begin{cases} a_k/2 & \text{si } k \text{ es par} \\ 3a_k + 1 & \text{si } k \text{ es impar} \end{cases} \quad (1)$$

hasta que en algún paso de la iteración tenemos $a_k = 1$.

¿Termina este proceso en un número finito de pasos para cualquier número A_0 ? (No se conoce la respuesta)

Ejercicio: Verificar para $a_0 = 27$

Encontrar...

Cosas que son difíciles de encontrar...

- ecuaciones lineales y no lineales
- máximos o mínimos

Linealidad

En matemática una función lineal es cualquier función f que satisface (principio de superposición):

$$f(x + y) = f(x) + f(y)$$

$$f(ax) = af(x)$$

Por ejemplo, $f(x) = 2x$ es una función lineal

Una ecuación lineal es una ecuación de la forma $f(x) = b$, donde f es una función lineal y b es una constante.

Linealidad

Mientras que las ecuaciones lineales en general se pueden resolver fácilmente, las ecuaciones no lineales son más difíciles de resolver

Algoritmo para la fórmula de Baskara

```
print("Considere el polinomio  $ax^2+bx+c$ ")
a=float(input("Ingrese el coef de grado 2: "))
b=float(input("Ingrese el coef de grado 1: "))
c=float(input("Ingrese el coef de grado 0: "))

from math import sqrt
D=b**2-4*a*c
if D>0:
    x1=(-b+sqrt(D))/(2*a)
    x2=(-b-sqrt(D))/(2*a)
    print("Las soluciones son",x1,"y",x2)
elif D==0:
    x=-b/(2*a)
    print("La solución es",x)
else:
    print("No hay solución real")
```

Python

Cómo programar una computadora

- Hay muchos lenguajes de programación
- Sirven para distintas cosas

El lenguaje python

- Alto o bajo nivel?
- interpretado o compilado?
- general o específico?

Sintaxis de python

Es el conjunto de reglas que establecen lo que se puede y lo que no se puede escribir para que sea interpretado correctamente por el lenguaje.

Hay algunas herramientas que nos orientan sobre cómo conviene escribir en python para que todo funcione mejor, por ejemplo:

- Zen de python
- PEP8
- Es importante desarrollar la capacidad de programar con estilo

Variables en python

Para comenzar a escribir código necesitamos saber que existen:

- Variables y valores
- Tipos

Instalación de python

(ver video de la clase para demostración de cómo se instala en Windows)