

Universidad Autónoma de Madrid
Departamento de Ingeniería Informática

Algoritmia y Estructuras de Datos Avanzadas

Práctica 1 – Python Básico y Min Heaps

- Asistencia mínima de un **85%**
- 3 prácticas y un examen final de prácticas
- Práctica ser realizara por **parejas**
- Cálculo de la Nota final:

$$\text{Calificación} = 0.4 * \text{Prácticas} + 0.6 * \text{Teoría}$$

- Obligatorio obtener una nota **mayor o igual a 5 puntos**
- Entrega tardía (podrá haber penalizaciones)

- Python: ficheros con extensión **.py**
- Jupyter Notebook: ficheros con extensión **.ipynb**

Entornos virtuales:

Anaconda

Computer program



IDE



Visual Studio Code

Computer program



Extensions



Jupyter

v2023.8.1002501831

Microsoft microsoft.com

70,100,590

★★★★☆ (292)

Jupyter notebook support, interactive programming and computing that supports Intellisense, debugging and more.

Disable

Uninstall

Switch to Pre-Release Version



This extension is enabled globally.



Pylint

v2023.6.0

Private

Microsoft microsoft.com

864,091

★★★★☆ (7)

Linting support for python files using 'pylint'.

Disable

Uninstall

Switch to Pre-Release Version



This extension is enabled globally.

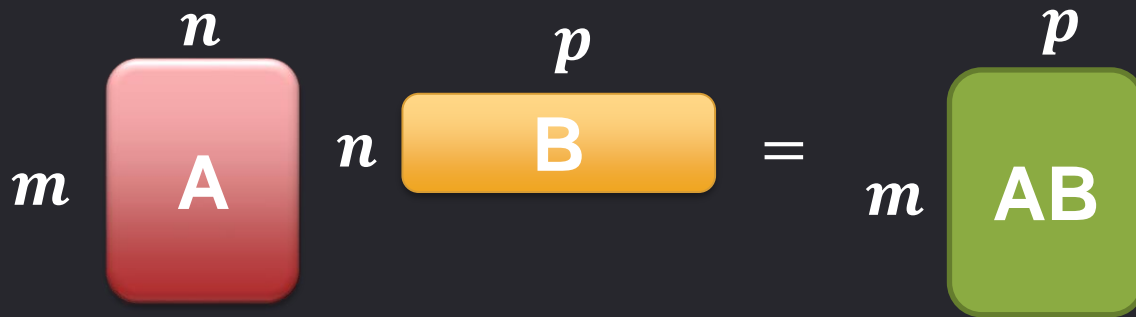
`src/python_examples`

- **Funciones y Módulos:** `functions_modules.ipynb`
- **Tipos de datos:** `data_types.ipynb`
- **Diccionarios:** `dictionaries.ipynb`
- **Sets:** `sets.ipynb`
- **Condicionales & Bucles:** `conditions_loops.ipynb`
- **Convertir listas:** `list_comprehensions.ipynb`
- **Zip:** `zip_argument_unpacking.ipynb`
- **Anotar el tipo de datos:** `type_annotations.ipynb`

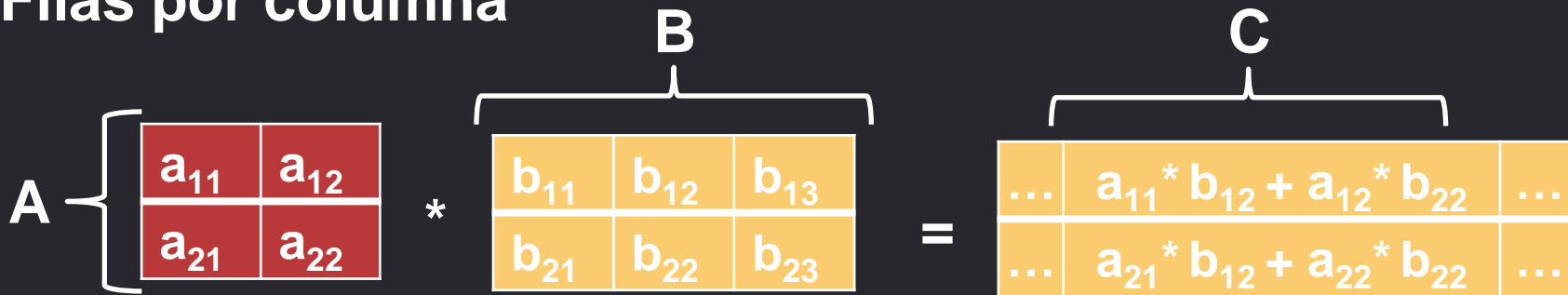
- Numpy: `src_numpy/numpy_functions.ipynb`
- Matplotlib: `src_visualization/matplot_lib.ipynb`

- ***ndarray***: contenedor multidimensional de elementos del mismo tipo y tamaño
- Número de dimensiones y elementos es definido por su ***shape***
- ***Ejemplos:***
 - `np.identity(3)`
 - `np.zeros(5)`
- **Enlaces útiles:**
 - <https://numpy.org/doc/stable/reference/arrays.ndarray.html>

- Multiplicación de matrices



- Filas por columna



- Este módulo proporciona una forma sencilla de medir el tiempo de ejecución en Python.

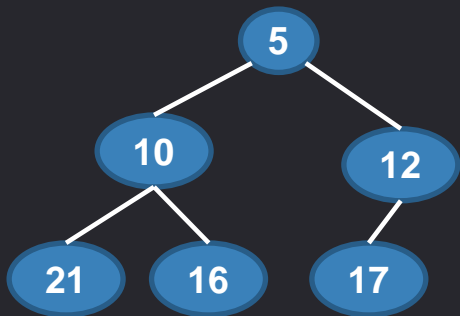
%timeit [-n<N> -r<R> [-t|-c] -q -p<P> -o]

- **-r:** Número de repeticiones
- **-o:** Retorna *TimeitResult*
- **-q:** No imprime el resultado
- **-n:** Ejecutar el código *N* veces
- **Enlaces útiles:**
 - <https://docs.python.org/3/library/timeit.html>
 - <https://ipython.org/ipython-doc/dev/interactive/magics.html#magic-timeit>

Práctica 1 – Python Básico – Búsqueda Binaria



- Heap es un árbol binario quasicompleto



- Heapify: comprobar que los hijos sean menores
- Insertar: hacer un push y ver si el padre es menor

- Heap se representa como una lista

- Formulas útiles:

i es la posición del elemento $Padre_i = (i - 1) / 2$

$$Hijo - Izquierdo_i = 2 * i + 1$$

$$Hijo - Derecho_i = 2 * i + 2$$

- **Python Básico**
- **ndarray & timeit**
- **Búsqueda Binaria**
- **Min Heaps**



Preguntas?