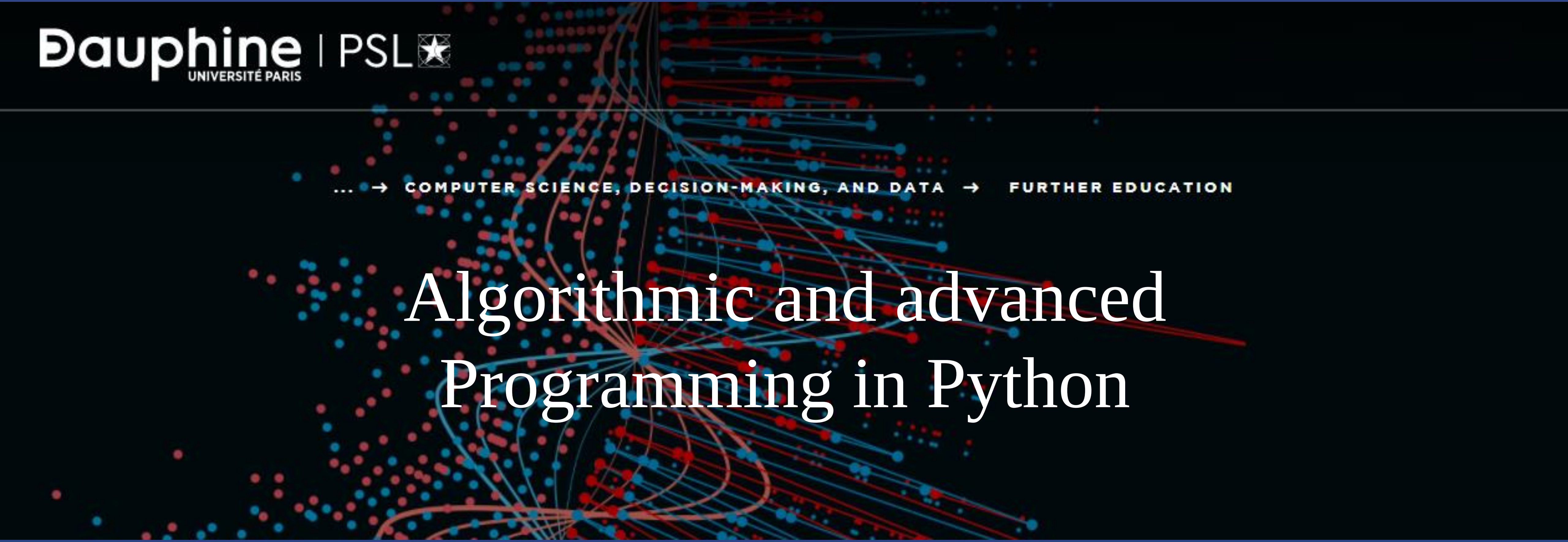
… → COMPUTER SCIENCE, DECISION-MAKING, AND DATA → FURTHER EDUCATION

Algorithmic and advanced Programming in Python

Remy Belmonte remy.belmonte@dauphine.eu

Lab 6

Problem 1: Check Binary Search Tree

- Give an algorithm to check a tree is a BST

Problem 2: Create a Balanced BST

- Write a Python program to create a Balanced Binary Search Tree (BST) using an array (given) elements where array elements are sorted in ascending order.

Problem 3: Binary Search Tree Search

- Write a Python program to find the closest value of a given target value in a given non-empty Binary Search Tree (BST) of unique values.

Problem 4: Delete a node in BST

- Write a Python program to delete a node with the given key in a given Binary search tree (BST).
- Note: Search for a node to remove. If the node is found, delete the node.

Problem 5: Conversion

- Write a Python program to convert a given array elements to a height balanced Binary Search Tree (BST).
- Note: The selection sort improves on the bubble sort by making only one exchange for every pass through the list.

Problem 6: kth smallest element in BST

- Write a Python program to find the kth smallest element in a given a binary search tree.

Problem 7: use BST to sort

- Write a Python program to sort a list of elements using Tree sort.

Problem 8: AVL Tree

- 1) Description:(Insertion In AVL)
  - Perform standard BST insert for w.
- 2) Starting from w, travel up and find the first unbalanced node.
  - Let z be the first unbalanced node, y be the child of z that comes on the path from w to z and x be the grandchild of z that comes on the path from w to z. 3) Re-balance the tree by performing appropriate rotations on the subtree rooted with z. There can be 4 possible cases that needs to be handled as x, y and z can be arranged in 4 ways. Following are the possible 4 arrangements:
    a) y is left child of z and x is left child of y (Left Left Case - LL)
    b) y is left child of z and x is right child of y (Left Right Case - LR)
    c) y is right child of z and x is right child of y (Right Right Case - RR)
    d) y is right child of z and x is left child of y (Right Left Case - RL)