TD8 – Graph Traversal

[Part 1: Graph Traversal 1](#_Toc600394515)

[Part 2: Shortest path in a graph: Dijkstra's algorithm. 1](#_Toc415109135)

For this lab, you are to use the class skeletons: *GraphTraversal* and some tests to check your codes: *GraphTraversalTest.* [*These classes were provided to you in the previous lab session.*](https://lms.univ-cotedazur.fr/2022/mod/folder/view.php?id=303729&forceview=1)

# Part 1: Graph Traversals (Facultatif)

You need to implement the graph traversal algorithms DFS and BFS in the *GraphTraversal* class.

# Part 2: Shortest path in a graph: Dijkstra's algorithm **(YOU HAVE TO IMPLEMENT IT)**

You need to implement Dijkstra's algorithm for finding the shortest path in a graph.

CoPilot or ChatGPT generate them quite well. Regardless of your choice for the implementation, make sure you understand what you are implementing.

For more information about the algorithm: <https://www.baeldung.com/cs/dijkstra-time-complexity>.

Please note that in the next session, you will be using your algorithm on a fairly large graph, so the complexity will be crucial.

Dans l’exemple ci-dessous le plus court chemin de A à E est :

A->B->D->E qui a pour distance : 6.5

Tandis que A->C->E a pour distance : 9,5

De C à E, il vaut mieux passer par D

