University of Sciences and Technology Houari Boumediene **2024 / 2025**
Computer Science Faculty, IASD Department **M1 SII**
Metaheuristics **2nd Semester**

Practical Work 2
Exact solving methods

Exact solving methods, also known as exhaustive search methods, explore the state space to construct solutions by systematically trying all possible combinations. Among the most popular exact solving algorithms, we find:

- **DFS (Depth First Search)**: Explores the deepest nodes first.
- **BFS (Breadth First Search)**: Explores the shallowest nodes first.

These algorithms are tree traversal algorithms. During tree construction, multiple branches or paths are accessible at each node. Each path from the root to a leaf represents a potential solution to the problem. The key difference between DFS and BFS lies in the order of node processing, DFS manages the list of candidate nodes as a stack (LIFO - Last In, First Out), whereas BFS processes the list of candidate nodes as a queue (FIFO - First In, First Out).

```
Procedure Search(G : Tree )

Input: Problem variables;
Output: BestSol : Node ;
Var
OPEN : Stack or Queue ; /* Depends on the algorithm used DFS or BFS */
CLOSE : List ; /* List of nodes already visited */
Begin
    d ←root(G) ;
    Insert d into OPEN ;
    While not Empty(OPEN) Do
        n = Open.node( ); /* Extract an element according to the FIFO or LIFO policy */
        Insert n into CLOSED ; /* Mark as already visited */
        IF FinalState(n) And Evaluation(G, n) > Evaluation(G, BestSol)
        Then Update(BestSol) ;   EndIF;
        ListChildren ← Successors(n) ;
        For each child e in ListChildren Do
            Insert e into OPEN ;
        Done ;
    Done ;
End;
```

To Do :

Implement the exact search algorithm Depth First Search (DFS) to solve the Knapsack Problem. Test your algorithm with different problem sizes. What do you observe?

Have Fun!