



جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم
قسم الانظمة الطبية
الذكية

Depth First Search

Lab: (3)

Subject: Artificial Intelligence

Class: Third

Lecturer: Dr. Maytham N. Meqdad





Python Program for DFS

<https://www.programiz.com/online-compiler/8Q1CTL1bKjHQ8>

```
# Graph represented as an adjacency list
graph = {
    'A': ['B', 'C'],
    'B': ['D', 'E'],
    'C': ['F'],
    'D': [],
    'E': ['F'],
    'F': []
}

# Iterative DFS using a stack
def dfs_iterative(graph, start):
    visited = set() # Track visited nodes to avoid cycles
    stack = [start] # Initialize stack with the starting node

    print("Iterative DFS:")
    while stack:
        node = stack.pop() # Pop a node from the stack
        if node not in visited:
            print(node, end=" ") # Process the node (e.g., print it)
            visited.add(node) # Mark it as visited

            # Add all unvisited neighbors to the stack
            for neighbor in reversed(graph[node]): # Reverse to maintain DFS order
                if neighbor not in visited:
                    stack.append(neighbor)
        print() # Newline for better readability

# Recursive DFS
def dfs_recursive(graph, node, visited=None):
    if visited is None:
        visited = set() # Initialize the set on the first call
        print("Recursive DFS:")

    visited.add(node)
    print(node, end=" ") # Process the node

    # Recursively visit all unvisited neighbors
    for neighbor in graph[node]:
        if neighbor not in visited:
            dfs_recursive(graph, neighbor, visited)
    if node == list(graph.keys())[0]: # Newline for better readability
        print()

# Running both DFS methods
dfs_iterative(graph, 'A')
dfs_recursive(graph, 'A')
```