**Research-Based Assignment on Uninformed and Informed Search Strategies**

**Part 1: Understanding Search Strategies (Conceptual Questions)**

**Task 1: Definitions and Characteristics**

1. Define **Uninformed Search** and provide two examples.
2. Define **Informed Search** and provide two.
3. Compare **Breadth-First Search (BFS)** and **Depth-First Search (DFS)** in terms of:
   * Completeness
   * Optimality
   * Time Complexity
   * Space Complexity
4. Why is *A Search*\* considered better than **Greedy Best-First Search (GBFS)**?

**Task 2: Problem-Solving Scenarios**

For each scenario below, suggest the most appropriate search strategy (from the allowed list) and justify your choice:

1. Finding the shortest path in a grid.
2. Solving an 8-puzzle problem with a heuristic (Research based).
3. Exploring all possible moves in a game tree without heuristic knowledge (Research based).

**Part 2: Implementation-Based Tasks**

**Task 3: Implementing BFS and DFS**

1. Write a Python program to implement **BFS** for finding the shortest path in a graph.
2. Modify the same program to implement **DFS** and compare the paths obtained.
3. Analyze which algorithm is more efficient for this problem and why (Research based).

**Task 4: Implementing Greedy Best-First Search (GBFS) and A**\*

1. Implement **Greedy Best-First Search** using a simple heuristic (e.g., Manhattan distance for a grid).
2. Extend the program to implement *A Search*\* with the same heuristic.
3. Compare the number of nodes explored by GBFS and A\* in a given maze.

**Part 3: Research and Analysis**

**Task 5: Case Study on Real-World Applications (Research based)**

Research and write a short report (200-300 words) on:

* **One real-world application of BFS/DFS** (e.g., web crawling, social networks).
* **One real-world application of A**\* (e.g., robotics, GPS navigation).
* Discuss why an informed search is preferred over an uninformed search in these cases.

**Task 6: Limitations and Trade-offs**

1. What are the main limitations of **Greedy Best-First Search**?
2. Under what conditions does **A**\* fail to find an optimal solution?
3. Why might **DFS** be impractical for large search spaces despite its low memory usage?

**Submission Guidelines**

* Include **code implementations** (Python files) for Tasks 3 and 4.
* Push the assignment file (along with Python files) to your GitHub repositories.