Quiz 2:

Tajbid Hasan

1. **Provide a brief introduction on how to use the 'Brute Force Solution(Dump)' to solve the Eight numbers problem.**

**The 'Brute Force Solution (Dump)' for the Eight Numbers problem involves systematically trying every possible combination of placing eight numbers (typically 0-7) on an 8x8 grid. This method does not use any sophisticated algorithms or optimizations; instead, it relies on generating all possible arrangements and then checking each one to see if it meets the criteria: no two numbers are in the same row, column, or diagonal.**

**In a typical implementation, you would use a series of nested loops, each iterating through the numbers 0-7. For each loop iteration, a number is placed in a row of the grid. After assigning numbers to all rows, the arrangement is checked for validity. If it meets the problem's constraints, it is considered a valid solution and can be printed or stored. This process repeats until all combinations have been exhausted. This brute force approach is straightforward but computationally expensive, as it involves checking a large number of combinations.**

1. **Provide a brief introduction on how to use the 'backtracking' to solve the Eight numbers problem.**

The C++ code uses a backtracking algorithm to solve the Eight Numbers problem. It involves a recursive function that attempts to place a number (1-8) at each position in an array, representing the grid. The algorithm checks for validity at each step: no two numbers are the same, and additional constraints specified in a helper array are met. If a number violates these rules, the algorithm backtracks and tries a different number. This process repeats until a valid arrangement is found for all positions, at which point the solution is printed. The algorithm efficiently explores the solution space, backtracking as soon as an invalid arrangement is detected.

1. **Provide a brief introduction on the pros and cons of "dump" and "backtracking."**

* ***Dump (Brute Force) Method:***
* Pros: Simple to implement; guarantees finding all possible solutions.
* Cons: Extremely inefficient for large problems; consumes significant time and computational resources.
* ***Backtracking Method:***
  + Pros: More efficient, reducing unnecessary computations; optimizes search process by eliminating wrong paths early.
  + Cons: Complex implementation; can still be time-consuming for vast solution spaces, though significantly less than brute force.

1. **(extra) What is the time complexity in question 1 and question 2?**
2. **Brute Force Solution (Dump):** Time complexity is O(nn)*O*(*nn*), where n*n* is the size of the grid (number of rows/columns). This reflects checking every possible arrangement for n*n* numbers.
3. **Backtracking:** Time complexity is generally less than O(n!)*O*(*n*!). It systematically explores possible placements but skips entire branches of the search tree when it encounters an invalid placement, thus reducing the number of configurations to examine.

**Question 1: Brute Force Solution (Dump)**

Time Complexity: O(88)*O*(88) - This method iterates through all possible combinations on an 8x8 grid, resulting in a highly exponential time complexity.

**Question 2: Backtracking Solution**

Time Complexity: Approximately O(8!)*O*(8!) - Backtracking significantly reduces the number of explored paths compared to brute force, but it still has an exponential time complexity in the worst case, typically factorial in nature for this type of problem.