[W6 - Data Structure & Algorithms]

Problem 1: Sorting Algorithms with Intermediate Results

Overview: Implement Selection sort and Bubble sort algorithms. Display the array's state after each iteration of the outer loop.

Selection Sort Algorithm:

1. **Initialize Variables**:
    - Start with the first element as the minimum.
2. **Iterate Through Array**:
    - For each iteration of the outer loop, find the minimum element from the unsorted part and swap it with the first element of the unsorted part.
3. **Display Intermediate Results**:
    - After each outer loop iteration, display the current state of the array.

Bubble Sort Algorithm:

1. **Initialize Variables**:
    - Start from the first element of the array.
2. **Iterate and Swap**:
    - In each iteration of the outer loop, bubble up the largest element to the end of the array by pairwise swapping.
3. **Display Intermediate Results**:
    - After each outer loop iteration, display the current state of the array.

Code

```
public class SortExample {
    public static void main(String[] args) {
        int[] array = {5, 1, 9, 6, 2};
        selectionSort(array);
        // bubbleSort(array);
    }

    private static void selectionSort(int[] arr) {
        int n = arr.length;
        for (int i = 0; i < n - 1; i++) {
            int minIndex = i;
            for (int j = i + 1; j < n; j++) {
                if (arr[j] < arr[minIndex]) {
                    minIndex = j;
                }
            }

            // Swap the found minimum element with the first element
            int temp = arr[minIndex];
            arr[minIndex] = arr[i];
            arr[i] = temp;

            // Print the current state of array
            printArray(arr);
        }
    }

    private static void bubbleSort(int[] arr) {
        int n = arr.length;
        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (arr[j] > arr[j + 1]) {
                    // Swap arr[j] and arr[j+1]
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
```

```
                }
            }

            // Print the current state of array
            printArray(arr);
        }
    }

    private static void printArray(int[] arr) {
        for (int value : arr) {
            System.out.print(value + " ");
        }
        System.out.println();
    }
}
```

Problem 2: Knapsack Problem

↳ *1 cell hidden*

Problem 3: 8-Queens with Pruning

↳ *1 cell hidden*