Pseudocode Outline: Topic 4

Sorting Algorithm Comparison

Purpose:

Compare different sorting algorithms by analyzing the average number of comparisons and data exchanges over 100 trials on randomly shuffled arrays.

```
BEGIN

  1. Prompt user to input an integer n (size of the array)

  2. Initialize statistics trackers for each sorting algorithm:
       ○ Selection Sort
       ○ Bubble Sort
       ○ Merge Sort
       ○ Quick Sort
       ○ Java Built-in Sort (as the optimized comparator)

  3. REPEAT the following steps 100 times:
     a. Create an array of size n containing values from 1 to n
     b. Shuffle the array randomly to simulate disorder
     c. FOR each sorting algorithm:
       ○ Clone the shuffled array
       ○ Sort the array using the algorithm
       ○ Track and record:
           ■ Number of comparisons mades
           ■ Number of data exchanges/swaps

  4. AFTER 100 trials:
       ○ FOR each sorting algorithm:
           ■ Calculate the average number of comparisons
           ■ Calculate the average number of data exchanges
           ■ Output the results in a formatted report

  5. END
```

Sorting Algorithms Used:

- Selection Sort
- Bubble Sort
- Merge Sort
- Quick Sort
- Java Arrays.sort() (Dual-Pivot Quick Sort used as better comparator)

Metrics Tracked:

- Comparisons
- Swaps / Data Exchanges

Notes:

- Java Arrays.sort() is used as the more efficient benchmark
- All arrays are re-randomized for fairness
- Cloning ensures identical input for each sorting method per trial