Task 2: Aggregated Threat Score Calculation

Revised Solution and Explanation

In this task, we need to calculate the **aggregated threat score** for departments based on their importance and individual threat scores. The goal is to test several scenarios using randomly generated data, such as departments having similar threat scores, high variance, or extreme values.

Changes and Improvements:

1. **Added type hints** for better clarity and maintainability.
2. Optimized the generation of random data to ensure values are within a specified range.
3. Ensured that the aggregated score is correctly normalized between 0 and 90.
4. Incorporated additional tests for varied cases, ensuring the code handles edge cases efficiently.

Explanation of Changes

1. **Function generate_random_data()**:
   - Added **type hints** for clarity (mean: int, variance: int, num_samples: int -> np.ndarray).
   - Used max(mean - variance, 0) and min(mean + variance + 1, 90) to ensure threat scores stay within the 0-90 range.
2. **Function calculate_aggregated_threat_score()**:
   - **Weighted threat scores**: The score is multiplied by the department's importance, reflecting the greater influence of more important departments.
   - **Normalized score**: The aggregated score is divided by the total importance and capped at 90, ensuring the result remains within the desired range.
3. **Unit Tests**:
   - **Various scenarios**: Tests cover cases where departments have similar threat scores, high variance, differing levels of importance, extreme values, and low threat scores.
   - **Assertions**: Ensured that the calculated threat score is within the expected range (0-90).

Test Cases Covered

- **Case 1**: Similar threat scores and equal importance across departments.
- **Case 2**: High variance in some departments' threat scores.
- **Case 3**: Departments with varying levels of importance.
- **Case 4**: Extreme threat scores (close to the upper bound of 90).
- **Case 5**: Low threat scores across all departments.