

**Batch:A3****Experiment Number: 3****Roll Number:16010423099****Name: Suryanshu Banerjee****Aim of the Experiment:** Implementation of Fenwick Tree operations

---

**Program/ Steps:**

```
class FenwickTree:
```

```
    def __init__(self, size):
```

```
        self.size = size
```

```
        self.tree = [0] * (size + 1) # 1-based indexing
```

```
    def update(self, index, delta):
```

```
        """Add delta to element at index"""
```

```
        index += 1 # Convert to 1-based indexing
```

```
        while index <= self.size:
```

```
            self.tree[index] += delta
```

```
            index += index & (-index) # Add least significant bit
```

```
    def query(self, index):
```

```
        """Get sum from index 0 to index"""
```

```
        index += 1 # Convert to 1-based indexing
```

```
        total = 0
```

```
        while index > 0:
```

```
            total += self.tree[index]
```

```
            index -= index & (-index) # Remove least significant bit
```

```
        return total
```

```

def range_query(self, left, right):
    """Get sum from left to right inclusive"""
    return self.query(right) - self.query(left - 1)

def build(self, array):
    """Build Fenwick tree from array"""
    for i, val in enumerate(array):
        self.update(i, val)

def test_fenwick_tree():
    # Test cases

    print("Test Case 1: Basic Operations")

    array = [2, 1, 4, 6, -1, 5, -32, 0, 1]

    ft = FenwickTree(len(array))

    ft.build(array)

    test_cases = [
        ("Query(0, 3)", ft.range_query(0, 3), "Sum of first 4 elements"),
        ("Query(2, 5)", ft.range_query(2, 5), "Sum of elements from index 2 to 5"),
        ("Query(0, 8)", ft.range_query(0, 8), "Sum of all elements")
    ]

    print("\nInitial Array:", array)

    for operation, result, description in test_cases:
        print(f"{operation}: {result} - {description}")

```

```
print("\nTest Case 2: Update Operations")

# Update value at index 4 to 10

old_val = array[4]

new_val = 10

ft.update(4, new_val - old_val)

array[4] = new_val


print(f"Updated index 4 from {old_val} to {new_val}")

print("New Array:", array)

print("Query(0, 8):", ft.range_query(0, 8))


if __name__ == "__main__":

    test_fenwick_tree()
```

**Output/Result:**

```
D:\PyCharm\MyProjects\contest995\.venv\Scripts\python.exe
Test Case 1: Basic Operations

Initial Array: [2, 1, 4, 6, -1, 5, -32, 0, 1]
Query(0, 3): 13 - Sum of first 4 elements
Query(2, 5): 14 - Sum of elements from index 2 to 5
Query(0, 8): -14 - Sum of all elements

Test Case 2: Update Operations
Updated index 4 from -1 to 10
New Array: [2, 1, 4, 6, 10, 5, -32, 0, 1]
Query(0, 8): -3

Process finished with exit code 0
```

---

**Post Lab Question-Answers:**

None.

---

**Outcomes:**

Understand the fundamental concepts for managing the data using different data structures such as lists, queues, trees etc.

---

**Conclusion (based on the Results and outcomes achieved):**

Successfully demonstrated Fenwick Tree with a program in Python.

---

**References:**

1. <https://www.hackerearth.com/practice/data-structures/advanced-data-ructures/segment-trees/tutorial/>
2. [https://cp-algorithms.com/data\\_structures/segment\\_tree.html](https://cp-algorithms.com/data_structures/segment_tree.html)