MUA

Maratones uniandes

TREES: FENWICK

ISIS 2801

Fenwick trees

Fenwick trees are used to calculate the associative operations over ranges in an array with minimal memory usage (`O(N)`).

Sometimes called
Binary Index Tree
(BIT)

A new data structure for cumulative frequency tables

Fenwick trees

Given an array

index	0	1	2	3	4	5	6	7	8	9	10
value	**2**	**3**	**4**	**1**	**7**	**2**	**8**	**7**	**4**	**1**	**5**

A (sum) Fenwick tree is

index	0	1	2	3	4	5	6	7	8	9	10
value	**2**	**5**	**4**	**10**	**7**	**9**	**8**	**34**	**4**	**5**	**5**

Fenwick trees

Given an array

index	0	1	2	3	4	5	6	7	8	9	10
value	**2**	**3**	**4**	**1**	**7**	**2**	**8**	**7**	**4**	**1**	**5**

A (sum) Fenwick tree is

$$T_i = \sum_{j=g(i)}^{i} A_j$$

Indexing function

The indexing function g for the Fenwick tree is such that

$$0 \leq g(i) \leq i$$

and we flip the least significant 1 bits of the binary representation of i

```
g(i) = i & (i+1)
```

Indexing function

The indexing function g for the Fenwick tree is such that

$$0 \leq g(i) \leq i$$

and we flip the least significant 1 bits of the binary representation of i

```
g(i) = i & (i+1)
```

Ex:

```
g(12) = g(1100₂) = 12
g(13) = g(1101₂) = 12
g(14) = g(1110₂) = 14
g(15) = g(1111₂) = 0
```

Indexing function

Need to find the function to get the indexes `j`

```
h(i) = i | (i+1)
```

Indexing function

0	1	2	3	4	5	6	7	8	9	10
2	**3**	**4**	**1**	**7**	**2**	**8**	**7**	**4**	**1**	**5**

Indexing function

0	1	2	3	4	5	6	7	8	9	10
2	**3**	**4**	**1**	**7**	**2**	**8**	**7**	**4**	**1**	**5**

$g(0) = g(0000_2) = 0$
$g(1) = g(0001_2) = 0$
$g(2) = g(0010_2) = 2$
$g(3) = g(0011_2) = 0$
$g(4) = g(0100_2) = 4$
$g(5) = g(0101_2) = 4$
$g(6) = g(0110_2) = 6$
$g(7) = g(0111_2) = 0$
$g(8) = g(1000_2) = 8$
$g(9) = g(1001_2) = 8$
$g(10) = g(1010_2) = 10$

Indexing function

0	1	2	3	4	5	6	7	8	9	10
2	**3**	**4**	**1**	**7**	**2**	**8**	**7**	**4**	**1**	**5**

$g(0) = g(0000_2) = 0$
$g(1) = g(0001_2) = 0$
$g(2) = g(0010_2) = 2$
$g(3) = g(0011_2) = 0$
$g(4) = g(0100_2) = 4$
$g(5) = g(0101_2) = 4$
$g(6) = g(0110_2) = 6$
$g(7) = g(0111_2) = 0$
$g(8) = g(1000_2) = 8$
$g(9) = g(1001_2) = 8$
$g(10) = g(1010_2) = 10$

sum(0, 0) = 2

Indexing function

0	1	2	3	4	5	6	7	8	9	10
2	**3**	**4**	**1**	**7**	**2**	**8**	**7**	**4**	**1**	**5**

$g(0) = g(0000_2) = 0$
$g(1) = g(0001_2) = 0$
$g(2) = g(0010_2) = 2$
$g(3) = g(0011_2) = 0$
$g(4) = g(0100_2) = 4$
$g(5) = g(0101_2) = 4$
$g(6) = g(0110_2) = 6$
$g(7) = g(0111_2) = 0$
$g(8) = g(1000_2) = 8$
$g(9) = g(1001_2) = 8$
$g(10) = g(1010_2) = 10$

sum(0, 0) = 2
sum(0, 1) = 5

Indexing function

0	1	2	3	4	5	6	7	8	9	10
2	**3**	**4**	**1**	**7**	**2**	**8**	**7**	**4**	**1**	**5**

$g(0) = g(0000_2) = 0$
$g(1) = g(0001_2) = 0$
$g(2) = g(0010_2) = 2$
$g(3) = g(0011_2) = 0$
$g(4) = g(0100_2) = 4$
$g(5) = g(0101_2) = 4$
$g(6) = g(0110_2) = 6$
$g(7) = g(0111_2) = 0$
$g(8) = g(1000_2) = 8$
$g(9) = g(1001_2) = 8$
$g(10) = g(1010_2) = 10$

sum(0, 0) = 2
sum(0, 1) = 5
sum(2, 2) = 4

Indexing function

0	1	2	3	4	5	6	7	8	9	10
2	**3**	**4**	**1**	**7**	**2**	**8**	**7**	**4**	**1**	**5**

$g(0) = g(0000_2) = 0$
$g(1) = g(0001_2) = 0$
$g(2) = g(0010_2) = 2$
$g(3) = g(0011_2) = 0$
$g(4) = g(0100_2) = 4$
$g(5) = g(0101_2) = 4$
$g(6) = g(0110_2) = 6$
$g(7) = g(0111_2) = 0$
$g(8) = g(1000_2) = 8$
$g(9) = g(1001_2) = 8$
$g(10) = g(1010_2) = 10$

sum(0, 0) = 2
sum(0, 1) = 5
sum(2, 2) = 4
sum(0, 3) = 10

Indexing function

0	1	2	3	4	5	6	7	8	9	10
2	**3**	**4**	**1**	**7**	**2**	**8**	**7**	**4**	**1**	**5**

```
g(0) = g(0000₂) = 0
g(1) = g(0001₂) = 0
g(2) = g(0010₂) = 2
g(3) = g(0011₂) = 0
g(4) = g(0100₂) = 4
g(5) = g(0101₂) = 4
g(6) = g(0110₂) = 6
g(7) = g(0111₂) = 0
g(8) = g(1000₂) = 8
g(9) = g(1001₂) = 8
g(10) = g(1010₂) = 10
```

```
sum(0, 0) = 2
sum(0, 1) = 5
sum(2, 2) = 4
sum(0, 3) = 10
sum(4, 4) = 7
```

Indexing function

0	1	2	3	4	5	6	7	8	9	10
2	**3**	**4**	**1**	**7**	**2**	**8**	**7**	**4**	**1**	**5**

```
g(0) = g(0000₂) = 0
g(1) = g(0001₂) = 0
g(2) = g(0010₂) = 2
g(3) = g(0011₂) = 0
g(4) = g(0100₂) = 4
g(5) = g(0101₂) = 4
g(6) = g(0110₂) = 6
g(7) = g(0111₂) = 0
g(8) = g(1000₂) = 8
g(9) = g(1001₂) = 8
g(10) = g(1010₂) = 10
```

```
sum(0, 0) = 2
sum(0, 1) = 5
sum(2, 2) = 4
sum(0, 3) = 10
sum(4, 4) = 7
sum(4, 5) = 9
```

Indexing function

0	1	2	3	4	5	6	7	8	9	10
2	**3**	**4**	**1**	**7**	**2**	**8**	**7**	**4**	**1**	**5**

$g(0) = g(0000_2) = 0$
$g(1) = g(0001_2) = 0$
$g(2) = g(0010_2) = 2$
$g(3) = g(0011_2) = 0$
$g(4) = g(0100_2) = 4$
$g(5) = g(0101_2) = 4$
$g(6) = g(0110_2) = 6$
$g(7) = g(0111_2) = 0$
$g(8) = g(1000_2) = 8$
$g(9) = g(1001_2) = 8$
$g(10) = g(1010_2) = 10$

sum(0, 0) = 2
sum(0, 1) = 5
sum(2, 2) = 4
sum(0, 3) = 10
sum(4, 4) = 7
sum(4, 5) = 9
sum(6, 6) = 8

Indexing function

0	1	2	3	4	5	6	7	8	9	10
2	**3**	**4**	**1**	**7**	**2**	**8**	**7**	**4**	**1**	**5**

$g(0) = g(0000_2) = 0$
$g(1) = g(0001_2) = 0$
$g(2) = g(0010_2) = 2$
$g(3) = g(0011_2) = 0$
$g(4) = g(0100_2) = 4$
$g(5) = g(0101_2) = 4$
$g(6) = g(0110_2) = 6$
$g(7) = g(0111_2) = 0$
$g(8) = g(1000_2) = 8$
$g(9) = g(1001_2) = 8$
$g(10) = g(1010_2) = 10$

sum(0, 0) = 2
sum(0, 1) = 5
sum(2, 2) = 4
sum(0, 3) = 10
sum(4, 4) = 7
sum(4, 5) = 9
sum(6, 6) = 8
sum(0, 7) = 34

Indexing function

0	1	2	3	4	5	6	7	8	9	10
2	**3**	**4**	**1**	**7**	**2**	**8**	**7**	**4**	**1**	**5**

$g(0) = g(0000_2) = 0$
$g(1) = g(0001_2) = 0$
$g(2) = g(0010_2) = 2$
$g(3) = g(0011_2) = 0$
$g(4) = g(0100_2) = 4$
$g(5) = g(0101_2) = 4$
$g(6) = g(0110_2) = 6$
$g(7) = g(0111_2) = 0$
$g(8) = g(1000_2) = 8$
$g(9) = g(1001_2) = 8$
$g(10) = g(1010_2) = 10$

sum(0, 0) = 2
sum(0, 1) = 5
sum(2, 2) = 4
sum(0, 3) = 10
sum(4, 4) = 7
sum(4, 5) = 9
sum(6, 6) = 8
sum(0, 7) = 34
sum(8, 8) = 4

Indexing function

0	1	2	3	4	5	6	7	8	9	10
2	**3**	**4**	**1**	**7**	**2**	**8**	**7**	**4**	**1**	**5**

$g(0) = g(0000_2) = 0$
$g(1) = g(0001_2) = 0$
$g(2) = g(0010_2) = 2$
$g(3) = g(0011_2) = 0$
$g(4) = g(0100_2) = 4$
$g(5) = g(0101_2) = 4$
$g(6) = g(0110_2) = 6$
$g(7) = g(0111_2) = 0$
$g(8) = g(1000_2) = 8$
$g(9) = g(1001_2) = 8$
$g(10) = g(1010_2) = 10$

sum(0, 0) = 2
sum(0, 1) = 5
sum(2, 2) = 4
sum(0, 3) = 10
sum(4, 4) = 7
sum(4, 5) = 9
sum(6, 6) = 8
sum(0, 7) = 34
sum(8, 8) = 4
sum(8, 9) = 5

Indexing function

0	1	2	3	4	5	6	7	8	9	10
2	**3**	**4**	**1**	**7**	**2**	**8**	**7**	**4**	**1**	**5**

```
g(0) = g(0000₂) = 0
g(1) = g(0001₂) = 0
g(2) = g(0010₂) = 2
g(3) = g(0011₂) = 0
g(4) = g(0100₂) = 4
g(5) = g(0101₂) = 4
g(6) = g(0110₂) = 6
g(7) = g(0111₂) = 0
g(8) = g(1000₂) = 8
g(9) = g(1001₂) = 8
g(10) = g(1010₂) = 10
```

```
sum(0, 0) = 2
sum(0, 1) = 5
sum(2, 2) = 4
sum(0, 3) = 10
sum(4, 4) = 7
sum(4, 5) = 9
sum(6, 6) = 8
sum(0, 7) = 34
sum(8, 8) = 4
sum(8, 9) = 5
sum(10, 10) = 5
```

Build

```
struct FenwickTree {
    vector<int> bit;  // binary indexed tree
    int n;

FenwickTree(vector<int> const &a) : FenwickTree(a.size()){
        for (int i = 0; i < n; i++) {
            bit[i] += a[i];
            int r = i | (i + 1);
            if (r < n) bit[r] += bit[i];
        }
    }
```

Query

```
int sum(int r) {
        int ret = 0;
        for (; r >= 0; r = (r & (r + 1)) - 1)
            ret += bit[r];
        return ret;
    }

int sum(int l, int r) {
    return sum(r) - sum(l - 1);
}
```

Update

```
void update(int n, int index, int val) {
        while (index <= n) {
            bit[index] += val;
            index += index & (-index);
        }
}
```