Fenwick Trees

Dr. Mattox Beckman

University of Illinois at Urbana-Champaign
Department of Computer Science

Objectives

Your Objectives:

- Describe and implement a Fenwick Tree
- Compare a Fenwick Tree to a Segment Tree

Motivating Example

Exam scores = $\{2, 4, 5, 5, 6, 6, 6, 7, 7, 8, 9\}$

Motivating Example

Exam scores = $\{2, 4, 5, 5, 6, 6, 6, 7, 7, 8, 9\}$

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

Motivating Example

Exam scores = $\{2, 4, 5, 5, 6, 6, 6, 7, 7, 8, 9\}$

index	1	2	3	4	5	6	7	8	9	10
value	0	1	0	1	2	3	2	1	1	0
cumulative	0	1	1	2	4	7	9	10	11	11

Motivating Example

Exam scores = $\{2, \mathbf{3}, 4, 5, 5, 6, 6, 6, 7, 7, 8, 9\}$

index	1	2	3	4	5	6	7	8	9	10
value	0	1	**1**	1	2	3	2	1	1	0
cumulative	0	1	**2**	**3**	**5**	**8**	**10**	**11**	**12**	**12**

Fenwick Tree

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

Fenwick Tree

value	0	1	0	1	2	3	2	1	1	0
index	1	2	3	4	5	6	7	8	9	10
index	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010

Fenwick Tree

ft 1	0		0		2		2		1	
value	0	1	0	1	2	3	2	1	1	0
index	1	2	3	4	5	6	7	8	9	10
index	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010

Fenwick Tree

ft 2		**1**				**5**				**1**
ft 1	0		0		2		2		1	
value	**0**	**1**	0	1	**2**	**3**	2	1	**1**	**0**
index	1	2	3	4	5	6	7	8	9	10
index	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010

Fenwick Tree

ft 3				**2**						
ft 2		1				5				1
ft 1	0		0		2		2		1	
value	**0**	**1**	**0**	**1**	2	3	2	1	1	0
index	1	2	3	4	5	6	7	8	9	10
index	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010

Fenwick Tree

ft 4								**10**		
ft 3				2						
ft 2		1				5				1
ft 1	0		0		2		2		1	
value	**0**	**1**	**0**	**1**	**2**	**3**	**2**	**1**	1	0
index	1	2	3	4	5	6	7	8	9	10
index	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010

Queries

ft 4								10		
ft 3				2						
ft 2		1				5				1
ft 1	0		0		2		2		1	
value	0	1	0	1	2	3	2	1	1	0
index	1	2	3	4	5	6	7	8	9	10
index	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010

- To Query sum for position *n*, first read *n*.
- Then, subtract the lowest order bit, and repeat until $n = 0$.

Queries

ft 4								10		
ft 3				2						
ft 2		1				5				1
ft 1	0		0		**2**		2		1	
value	0	1	0	1	2	3	2	1	1	0
index	1	2	3	4	**5**	6	7	8	9	10
index	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010

- To Query sum for position *n*, first read *n*.
- Then, subtract the lowest order bit, and repeat until $n = 0$.

- E.g.: 5 = 101 $\rightarrow$ 4 = 100 $\rightarrow$ 0
  - $ft(5) + ft(4) = 2 + 2 = 4$

Queries

ft 4								10		
ft 3				**2**						
ft 2		1				5				1
ft 1	0		0		**2**		2		1	
value	0	1	0	1	2	3	2	1	1	0
index	1	2	3	**4**	**5**	6	7	8	9	10
index	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010

- To Query sum for position *n*, first read *n*.
- Then, subtract the lowest order bit, and repeat until $n = 0$.

- E.g.: 5 = 101 $\rightarrow$ 4 = 100 $\rightarrow$ 0
  - $ft(5) + ft(4) = 2 + 2 = 4$

Queries

ft 4								10		
ft 3				2						
ft 2		1				5				**1**
ft 1	0		0		2		2		1	
value	0	1	0	1	2	3	2	1	1	0
index	1	2	3	4	5	6	7	8	9	**10**
index	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010

- To Query sum for position *n*, first read *n*.
- Then, subtract the lowest order bit, and repeat until $n = 0$.

- E.g.: 5 = 101 → 4 = 100 → 0
  - $ft(5) + ft(4) = 2 + 2 = 4$
- E.g.: 10 = 1010 → 8 = 1000 → 0
  - $ft(10) + ft(8) = 1 + 10 = 11$

Queries

ft 4								**10**		
ft 3				2						
ft 2		1				5				**1**
ft 1	0		0		2		2		1	
value	0	1	0	1	2	3	2	1	1	0
index	1	2	3	4	5	6	7	**8**	9	**10**
index	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010

- To Query sum for position *n*, first read *n*.
- Then, subtract the lowest order bit, and repeat until $n = 0$.

- E.g.: 5 = 101 $\rightarrow$ 4 = 100 $\rightarrow$ 0
  - $ft(5) + ft(4) = 2 + 2 = 4$
- E.g.: 10 = 1010 $\rightarrow$ 8 = 1000 $\rightarrow$ 0
  - $ft(10) + ft(8) = 1 + 10 = 11$

Queries

ft 4								10		
ft 3				2						
ft 2		1				5				1
ft 1	0		0		2		2		1	
value	0	1	0	1	2	3	2	1	1	0
index	1	2	3	4	5	6	7	8	9	10
index	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010

- To Query sum for position *n*, first read *n*.
- Then, subtract the lowest order bit, and repeat until $n = 0$.

- E.g.: 5 = 101 $\rightarrow$ 4 = 100 $\rightarrow$ 0
  - $ft(5) + ft(4) = 2 + 2 = 4$
- E.g.: 10 = 1010 $\rightarrow$ 8 = 1000 $\rightarrow$ 0
  - $ft(10) + ft(8) = 1 + 10 = 11$

- Update 5: Visit 5=101 $\rightarrow$ 6=110 $\rightarrow$ 8=1000

Queries

ft 4								10		
ft 3				2						
ft 2		1				5				1
ft 1	0		0		2		2		1	
value	0	1	0	1	2	3	2	1	1	0
index	1	2	3	4	5	6	7	8	9	10
index	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010

- To Query sum for position *n*, first read *n*.
- Then, subtract the lowest order bit, and repeat until $n = 0$.

- E.g.: 5 = 101 → 4 = 100 → 0
  - $ft(5) + ft(4) = 2 + 2 = 4$
- E.g.: 10 = 1010 → 8 = 1000 → 0
  - $ft(10) + ft(8) = 1 + 10 = 11$

- Update 5: Visit 5=101 → 6=110 → 8=1000
- `LSOne(n) = n & -n`