Segment Trees

Dr. Mattox Beckman

University of Illinois at Urbana-Champaign
Department of Computer Science

Objectives

Your Objectives:

- Write the code for a segment tree
- Solve the Range Minimum Query problem
- Solve the Range Sum Query problem

Range Minimum Query

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

Segment Tree

The array:

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

The segment tree:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
[0,7]	[0,3]	[4,7]	[0,1]	[2,3]	[4,5]	[6,7]	[0,0]	[1,1]	[2,2]	[3,3]	[4,4]	[5,5]	[6,6]	[7,7]
5	3	5	1	3	5	7	0	1	2	3	4	5	6	7

Queries

The array:

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

The segment tree:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
[0,7]	[0,3]	[4,7]	[0,1]	[2,3]	[4,5]	[6,7]	[0,0]	[1,1]	[2,2]	[3,3]	[4,4]	[5,5]	[6,6]	[7,7]
5	3	5	1	3	5	7	0	1	2	3	4	5	6	7

- A query like [0,7] or [4,5] can be answered in one lookup.
- A query like [1,4] needs 3 lookups: [1,1], [2,3], [4,4].

Query

```
0   int rmq(int p, int L, int R, int i, int j) {
1     if (i > R || j < L) return -1; // current segment outside query ran
2     if (L >= i && R <= j) return st[p];
3     // compute the min position in the left and right part of the inter
4     int p1 = rmq(left(p) , L         , (L+R)/2, i, j);
5     int p2 = rmq(right(p), (L+R)/2+1, R      , i, j);
6     if (p1 == -1) return p2;
7     // if we try to access segment outside query
8     if (p2 == -1) return p1;
9     // same as above
10    return (A[p1] <= A[p2]) ? p1 : p2;
11  }
```

Building

```
0   void build(int p, int L, int R) {
1     if (L == R) // as L == R, either one is fine
2        st[p] = L; // store the index
3     else {
4       // recursively compute the values
5       build(left(p) , L               , (L + R) / 2);
6       build(right(p), (L + R) / 2 + 1, R           );
7       int p1 = st[left(p)], p2 = st[right(p)];
8       st[p] = (A[p1] <= A[p2]) ? p1 : p2;
9     } }
```

Dynamic Updates

Suppose we update element 2 from 7 to 1...

0	1	2	3	4	5	6	7
8	6	$7 \rightarrow 1$	5	3	0	9	5

The resulting segment tree:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
[0,7]	[0,3]	[4,7]	[0,1]	[2,3]	[4,5]	[6,7]	[0,0]	[1,1]	[2,2]	[3,3]	[4,4]	[5,5]	[6,6]	[7,7]
5	$3 \rightarrow 2$	5	1	$3 \rightarrow 2$	5	7	0	1	2	3	4	5	6	7

Range Sum

The array:

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

The segment tree:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
[0,7]	[0,3]	[4,7]	[0,1]	[2,3]	[4,5]	[6,7]	[0,0]	[1,1]	[2,2]	[3,3]	[4,4]	[5,5]	[6,6]	[7,7]
43	26	17	14	12	3	14	8	6	7	5	3	0	9	5