ECE 5960-023/6960-025 – Advanced Programming for Computer Design Problems

In-class Practice 12 (due 2020/4/16 to tsung-wei.huang@utah.edu)

1. Construct the following sparse table for an array of eight element (mark in red) in the minimum domain.

	1 **(2)**	2 **(4)**	3 **(5)**	4 **(1)**	5 **(-5)**	6 **(8)**	7 **(11)**	8 **(6)**
0								
1								
2								
3								

2. Layout the binary representation of the following three operations:

i. 5 & -5
ii. 4 & -4
iii. 18 & -18

3. Perform the iterative lowbit operation "$a_{i+1}=a_i$ & $-a_i$" on the number 21 (10101'b) until you reach the power of two. Show your progress.

4. Draw the binary indexed tree on the array A[20] = {1, 4, -9, 1, -2, -4, 5, 8, 9, 2, 3, 4, 5, 12, 21, 16, 8, -5, -3, 9} for M[i], where M represents the maximum value in the corresponding range.

Name: uid:

5. Finish `rq.cpp` by implementing the two function `find_sum_st` and `find_sum_bit` that answer the query of the sum value over a range using sparse table method and binary indexed tree method. Measure the runtime each function tasks to complete each testcase, totally 4. Complete the following table with your runtime data.

	test1.txt	test2.txt	test3.txt	test4.txt
Naïve method				
Sparse Table				
Binary Indexed Tree				

In detail, what is the time it takes to construct the sparse table and binary indexed tree, respectively? You can use `std::chrono::steady_clock::now` facility to measure the runtime in code.

Which method you prefer?

6. Watch the video "Quantifying Accidental Complexity: An Empirical Look at Teaching and Using C++" presented by Herb Sutter: https://www.youtube.com/watch?v=qx22oxlQmKc. Give your comment.