Justin Mendiguarin

Professor Ouyang

CSC 130

9/30/15

Assignment 1 - Maximum Subsequence Product

**Analysis:**

T(maxSubseqProduct, N) = 1 + 2T(maxSubseqProduct, ) + 2( ) // Constant(1)

= 1 + 2T(maxSubseqProduct, ) + N = + 1

* Throw away leading constants

= 1 + log(N)

T(maxSubseqProduct, N) = N + N\*log(N)

= O(N\*log(N))

maxSubseqProduct(**int**[] ARRAY, **int** start, **int** end) {

int leftMaxProduct = *maxSubseqProduct*(ARRAY, start, center); // T(maxSubseqProduct, N/2)

int rightMaxProduct = *maxSubseqProduct*(ARRAY, center + 1, end); // T(maxSubseqProduct, N/2)

**int** leftProductofCrossinMiddle = 0, maxLeftCenterProduct = 0, minLeftCenterProduct = 0; // Constant

leftProductofCrossinMiddle = maxLeftCenterProduct = minLeftCenterProduct = ARRAY[center]; // Constant

//FOR LOOP for the Left Max Center Product

**for** (**int** i = center - 1; i >= start; i--) { // (N/2) for first FOR loop.

}

**int** rightProductofCrossinMiddle = 0, maxRightCenterProduct = 0, minRightCenterProduct = 0; // Constant

rightProductofCrossinMiddle = maxRightCenterProduct = minRightCenterProduct = ARRAY[center + 1]; // Constant

//FOR LOOP for the Right Max Center Product

**for** (**int** i = center + 2; i <= end; i++) { // (N/2) for second FOR loop

}

}

**public** **static** **int** maxSubseqProduct(**int**[] ARRAY, **int** start, **int** end) {

//Base Case

**if** (start == end)

**return** (ARRAY[start]);

**int** center = (start + end) / 2;

**int** leftMaxProduct = *maxSubseqProduct*(ARRAY, start, center); // T(maxSubseqProduct, N/2)

**int** rightMaxProduct = *maxSubseqProduct*(ARRAY, center + 1, end); // T(maxSubseqProduct, N/2)

// Constant

**int** leftProductofCrossinMiddle = 0, maxLeftCenterProduct = 0, minLeftCenterProduct = 0; // Constant

leftProductofCrossinMiddle = maxLeftCenterProduct = minLeftCenterProduct = ARRAY[center]; // Constant

**int** maxLeftSequenceStart, minLeftSequenceStart;

maxLeftSequenceStart = minLeftSequenceStart = center;

// (N/2) for first For loop.

//FOR LOOP for the Left Max Center Product

**for** (**int** i = center - 1; i >= start; i--) {

leftProductofCrossinMiddle \*= ARRAY[i];

**if** (leftProductofCrossinMiddle > maxLeftCenterProduct) {

maxLeftCenterProduct = leftProductofCrossinMiddle;

maxLeftSequenceStart = i;

}

**else** **if** (leftProductofCrossinMiddle < minLeftCenterProduct) {

minLeftCenterProduct = leftProductofCrossinMiddle;

minLeftSequenceStart = i;

}

}

// Constant

**int** rightProductofCrossinMiddle = 0, maxRightCenterProduct = 0, minRightCenterProduct = 0; // Constant

rightProductofCrossinMiddle = maxRightCenterProduct = minRightCenterProduct = ARRAY[center + 1]; //Constant

**int** maxRightSequenceEnd, minRightSequenceEnd;

maxRightSequenceEnd = minRightSequenceEnd = center + 1;

// (N/2) for second For loop.

//FOR LOOP for the Right Max Center Product

**for** (**int** i = center + 2; i <= end; i++) {

rightProductofCrossinMiddle \*= ARRAY[i];

**if** (rightProductofCrossinMiddle > maxRightCenterProduct) {

maxRightCenterProduct = rightProductofCrossinMiddle;

maxRightSequenceEnd = i;

}

**else** **if** (rightProductofCrossinMiddle < minRightCenterProduct) {

minRightCenterProduct = rightProductofCrossinMiddle;

minRightSequenceEnd = i;

}

}

// Combine the solution from the left and right max center product to arrive at a

// Solution for the max center product.

**int** maxCenterProduct = maxLeftCenterProduct \* maxRightCenterProduct;

**int** minCenterProduct = minLeftCenterProduct \* minRightCenterProduct;

// Compare the the four values of which is the maximum product

**int** max = *getMax*(leftMaxProduct, rightMaxProduct, maxCenterProduct,

minCenterProduct);

**if** (maxCenterProduct >= max) {

start = maxLeftSequenceStart;

end = maxRightSequenceEnd;

}

**if** (minCenterProduct >= max) {

start = minLeftSequenceStart;

end = minRightSequenceEnd;

}

System.***out***.println("Start: " + start + " End: " + end);

System.***out***.print("Maximum Subsequence: [");

**for** (**int** i = start; i <= end; i++) {

System.***out***.print("" + ARRAY[i] + ",");

}

System.***out***.println("]");

System.***out***.println("max: " + max);

System.***out***.println();

**return** max;

}