**Recursive Implementaion2**

**Algorithm iterative\_candles(candles: array of integers, n: integer) returns MaxheightCount**

**{**

**// Initialize variables**

**result := {.height = candles[0], .count = 0};**

**max\_height := candles[0];**

**count := 0;**

**// Iterate through the candles**

**for i := 0 to n - 1 do**

**{**

**// Check if the current candle height is greater than the current max height**

**if candles[i] > max\_height then**

**{**

**max\_height := candles[i]; // Update max height**

**count := 1; // Reset count to 1**

**}**

**// Check if the current candle height is equal to the current max height**

**else if candles[i] = max\_height then**

**{**

**count := count + 1; // Increment count**

**}**

**}**

**// Update result with max height and count**

**result.height := max\_height;**

**result.count := count;**

**return result; // Return the result**

**}**

**Algorithm recursive\_birthdayCandles2(candles: array of integers, n: integer) returns integer**

**{**

**// Check if there are no candles**

**if n <= 0 then**

**{**

**return 0; // Return 0 if there are no candles**

**}**

**// Call the iterative\_candles function to find max height count**

**max\_height\_count := iterative\_candles(candles, n);**

**return max\_height\_count.count; // Return the count of candles with max height**

**}**