4- Birthday Cake Candles

Recursive Algorithm Pseudo-code

```
function birthdaycakecandles(arr)
{
     return helper(arr, 0, none, 0)

     function helper(arr, i, current_max, count)
     {
          if (i == length(arr)):
               return count

          if (current_max == null  or  arr[i] > current_max):
               return helper(arr, i + 1, arr[i], 1)
          else if (arr[i] == current_max):
               return helper(arr, i + 1, current_max, count + 1)
          else
               return helper(arr, i + 1, current_max, count)
     }
}
```

Algorithm Analysis

Recurrence Relation:

$$T(n) = T(n-1) + c$$

$$T(1) = 1$$

Solving the Recurrence:

$$T(n) = T(n-1) + c$$

$$T(n-1) = T([n-1]-1) + c$$

$$T(n-1) = T(n-2) + 2c$$

$$T(n-2) = T([n-2]-1) + c$$

$$T(n-2) = T(n-3) + 3c$$

$$T(n) = T(n-k) + kc$$

$$n - k = 1$$

$$k = n - 1$$

$$T(n) = T(n-[n-1]) + (n-1)c$$

$$T(n) = T(1) + nc - c$$

$$T(n) = (1-c) + nc$$

$$T(n) \approx n$$

$$T(n) = O(n)$$

Result:

Best Case : $\Omega(n)$ Worst Case: $O(n)$ Average Case: $\theta(n)$