**Time Complexity**

#include <stdio.h>

#include "RecursiveCandels2.h"

MaxheightCount recursive\_birthdayCakeCandles2(int\* candles, int n) {

MaxheightCount max\_height\_count ={.height = 0,.count = 0};------🡪O(1)

int\* stack = (int\*)malloc(n \* sizeof(int));

int top = -1;

for (int i = 0; i < n; i++) { ------🡪O(n)

while (top >= 0 && candles[stack[top]] < candles[i]) {------🡪O(n)

top--;

}

if (top == -1 || candles[stack[top]] == candles[i]) {------🡪O(1)

top++;

stack[top] = i;

}

}

int max\_height = candles[stack[0]];

int count = 1;

for (int i = 1; i <= top; i++) {------🡪O(n)

if (candles[stack[i]] == max\_height) {------🡪O(1)

count++;

}

else {------🡪O(1)

break;

}

}

max\_height\_count.height = max\_height;

max\_height\_count.count = count;

free(stack);

return max\_height\_count;

}

**T(n) =** *O*(1)+*O*(*n*)+*O*(*n*)+*O*(1)+*O*(*n*)+*O*(1)+*O*(1)+*O*(1) = O(n)