

Punto 1

```
def arrayMax_aux(arr, i, max):  
    if i == len(arr):  
        return max  
    else:  
        if arr[i] > max:  
            max = arr[i]  
        return arrayMax_aux(arr, i + 1, max)
```

#constante
#retorna variable

#variable
#variable
#retorna T(n-1)

#retorna variable max cuando $c_2 == n$
#retorna $T(n-1)$ cuando $C_2 < n$

$t(n) = t(n-1) + k$
 $t(n) = c_1 + k \cdot n$
 $t(n) = O(k \cdot n + c_1)$ por definición de O
 $t(n) = O(k \cdot n)$ por regla de la suma
 $t(n) = O(n)$ por regla del producto

Punto 2

```
def groupSum_aux(list, start, target):  
    if start >= len(list):  
        return target == 0  
    return groupSum_aux(list, start + 1, target - list[start]) \  
        or groupSum_aux(list, start + 1, target)
```

#constante
#constante
#constante + $t(n-1)$
#constante + $t(n-1)$

#retorna c_1 cuando $n = 1$
#retorna $2 * t(n-1)$ cuando $n > 1$

$t(n) = t(n-1) + t(n-1)$
 $t(n) = c_1 * 2^{n-1}$
 $t(n) = O(c_1 * 2^{n-1})$ por definición de O
 $t(n) = O(2^{n-1})$ por regla del producto
 $t(n) = O(2^n)$ por regla de la suma