Algorithmics	Student information	Date	Number of session
	UO: 283069	2/03/2022	3.1
	Surname: Carvajal Aza		
	Name: Guillermo Dylan		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. Basic Recursive Models

Division 1

Since we know that a = 1, b = 3 and k = 1, taking into account that we are using a recursive method by division and a < b^k we conclude that the complexity is O(n^1) = O(n).

Division 2

For this method, a = 2, b = 2 and k = 1, therefore a = b^k and the complexity is O(n*logn).

Division 3

On this method, we have that a = 2, b = 2 and k = 0, then the complexity is calculated as O(n^logb(a)), since a > b^k, and the complexity is: O(n^log2(2)) that is the same as O(n).

Division 4

The complexity for this algorithm is stated on the wording for this session, its O(n^2), and in order to achieve it we would need that:

a = 4 b >= 2 k = 2

Since then a < b^k and the complexity is calculated with O(n^k).

Although this was the method I applied for getting the complexity, it is also true that we could achieve it as well by having a = 4, b = 2 and k <= 1, since then the complexity would be calculated with O(n^(logb(a)), to which if we apply the values stated is the same as O(n^2), but since we are also told that the number of sub problems must be 4 we cannot use this solution and must apply the first mentioned.

Subtraction 1

Now we have that the methods use recursion by subtraction instead of by division.

For this method: a = 1, b = 1 and k = 0, therefore a = 1

Then the complexity is calculated as O(n^(k+1)) which is O(n).

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Algorithmics	Student information	Date	Number of session
	UO: 283069	2/03/2022	3.1
	Surname: Carvajal Aza		
	Name: Guillermo Dylan		

Subtraction 2

For the second method using recursion by subtraction we have that a = 1, b = 1 and k = 1, and again the complexity is calculated as O(n^(k+1)), since a is still equals to 1.

Therefore the complexity is O(n^2).

Subtraction 3

This method has that a = 2, b = 1 and k = 0, so, as a > 1 we have that the complexity is:

O(a^(n/b) = O(2^(n/1)) = O(2 ^ n)

Subtraction 4

The complexity for this method must be O(3^(n/2)), and in order to obtain it we are going to use:

a = 3 b = 2 k = 0

By doing this we obtain a complexity of O(n^3(n/2)), since a > 1 and the formula for recursion by subtraction on this case is O(a^(n/b)).