Algorithmics	Student information	Date	Number of session
	UO: 302165	27/02/2025	3
	Surname: Uña Garcia		
	Name: Lucas		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. [D&C BY SUBSTRACTION]

Substraction1:

a = 1, b = 1, k = 0

Complexity:

$O(n^{k+1})$ => $O(n)$

I cannot prove if the results fit or not with the theoretical complexity because the times are not accurate (below 50ms) and then the stack overflow exception rises.

Substraction2:

a = 1, b = 1, k = 1

Complexity:

$O(n^{k+1})$ => $O(n^2)$

The results obtained fits with the theoretical complexity. Although for values of n < 2048 are not accurate, for higher values until the stack overflow exception we can appreciate that the time grows by a factor of two, which is the square of the increment of the size (2^2).

Substraction3:

a = 2, b = 1, k = 0

Complexity:

$O(a^{n/b})$ => $O(2^n)$

The results obtained are the ones expected according to the theoretical complexity, because as n increases one by one, the time increases exponentially (the new time is the previously obtained times 2, as $2^{n+1} = 2^n * 2$).

Escuela de Ingeniería Informática

Algorithmics	Student information	Date	Number of session
	UO: 302165	27/02/2025	3
	Surname: Uña Garcia		
	Name: Lucas		

- The operation is aborted not because the time but the memory occupied. As for each iteration a variable is created, the stack has less and less space until it is full occupied.
- N1=20 → t1 = 41

  N2=80 → t2 = ?

  t2 = $2^{N2}/2^{N1}$ * t1 = 2^{N2-N1} * t1 = 2^{60} * t1 = 47.269.781.688.880.726.016ms * 1s/1000ms * 1h/3600s * 1day/24h * 1year/365.25days = 1.497.888.993,1 years

Substraction4	
n	time(ms)
100	1,425
200	10,65
400	82,15
800	650
1600	5013
3200	40532

Substraction5	
n	time(ms)
32	1098
34	3291
36	9873
38	29285

- N1=32 → t1 = 1098

  N2=80 → t2 = ?

  t2 = $3^{N2/2}/3^{N1/2}$ * t1 = $3^{N2/2 - N1/2}$ * t1 = 3^{24} * t1 = 310.107.631.056.138ms* 1s/1000ms * 1h/3600s * 1day/24h * 1year/365.25days = 9.826,72years

Algorithmics	Student information	Date	Number of session
	UO: 302165	27/02/2025	3
	Surname: Uña Garcia		
	Name: Lucas		

Activity 2. [D&C BY DIVISION]

Division1:

a = 1, b = 3, k = 1

Complexity:

$O(n^k)$ => $O(n)$

The results obtained fits with the theoretical complexity because as the size increases by a factor of 2, same happens with the time, so the time increases linearly.

Division2:

a = 2, b = 2, k = 1

Complexity:

$O(n^k*\log(n))$ => $O(n*\log(n))$

According to the theoretical complexity, the results obtained are the ones expected. The size increments by a factor of 2 but the time is increased by a bit more than 2 which reflects that multiplication by log(n) of the theoretical calculus.

Division3:

a = 2, b = 2, k = 0

Complexity:

$O(n^{\log_b a})$ => $O(n)$

We are in the same case that with Division1 and the results obtained also fits the constraints required (has a linear time increment according to the size increment).

Algorithmics	Student information	Date	Number of session
	UO: 302165	27/02/2025	3
	Surname: Uña Garcia		
	Name: Lucas		

Division4:

Division4	
n	time(ms)
1000	7,9
2000	29,2
4000	116
8000	465,2
16000	1820,5
32000	7293
64000	28918
128000	Oot

Division5:

Division5	
n	time(ms)
1000	23,4
2000	92
4000	369,5
8000	1502,5
16000	5997
32000	23637
64000	Oot

Activity 3. [TWO BASIC PRINCIPLES]

Fibonacci1:

Complexity:

O(n)

Algorithmics	Student information	Date	Number of session
	UO: 302165	27/02/2025	3
	Surname: Uña Garcia		
	Name: Lucas		

Fibonacci2:

Complexity:

O(n)

Fibonacci3:

a = 1, b = 1, k = 0

Complexity:

$O(n^{k+1})$ => O(n)

Fibonacci4:

a = 2, b = 2 or 1, k = 0

Complexity:

$O(a^{n/b})$ => $<O(2^n)$

n	timeF1	timeF2	timeF3	timeF4
10	0,000088	0,000114	0,000191	0,00243
11	0,000092	0,000119	0,000222	0,00381
12	0,000093	0,000125	0,000236	0,00614
13	0,000097	0,000129	0,000253	0,00986
14	0,000101	0,000134	0,000268	0,01592
15	0,000105	0,000141	0,000284	0,02575
16	0,000109	0,000148	0,000312	0,04192
17	0,000112	0,000155	0,000318	0,06852
18	0,000116	0,000161	0,000335	0,105
19	0,000124	0,000167	0,000348	0,171
20	0,000125	0,000175	0,000364	0,276
21	0,000131	0,000183	0,000381	0,444
22	0,000134	0,000191	0,000397	0,745
23	0,00014	0,000201	0,000412	1,188
24	0,00014	0,000209	0,000434	1,909
25	0,000147	0,000213	0,000449	3,071

As can be seen all the times fits with its theoretical complexity. As can be seen the first method, even if they have the same complexity, it's a bit better due to the simplicity of its implementation.

VectorSum1:

Complexity:

O(n)

VectorSum2:

a = 1, b = 1, k = 0

Complexity:

$O(n^{k+1}) \Rightarrow O(n)$

VectorSum3:

a = 2, b = 2, k = 0

Complexity:

$O(n^{\log_b a}) \Rightarrow O(n)$

n	timeSum1	timeSum2	timeSum3
3	0,00005	0,000074	0,000085
6	0,000067	0,000124	0,000163
12	0,000093	0,000235	0,000345
24	0,000141	0,000428	0,000709
48	0,000237	0,000841	0,001456
96	0,00043	0,001624	0,002876
192	0,000793	0,00318	0,00581
384	0,001497	0,006361	0,011583
768	0,002956	0,01274	0,02324
1536	0,00589	0,02513	0,0465
3072	0,01169	0,04912	0,09336

Here happens same as in the previous example with the Fibonacci series.

Algorithmics	Student information	Date	Number of session
	UO: 302165	27/02/2025	3
	Surname: Uña Garcia		
	Name: Lucas		

Activity 4. [PETANQUE CHAMPIONSHIP]

Algorithm:

a = 2, b = 2, k = 1

Complexity:

O(n^klogn) => O(nlogn)

n	tCalendar
2	0,00067
4	0,00305
8	0,0118
16	0,0346
32	0,1435
64	0,54
128	2,22
256	8,9
512	37,04

The results obtained does not match the theoretical complexity so I think I have made an error in the calculation. The results corresponds with a quadratic complexity.

Escuela de Ingeniería Informática