Algorithmics	Student information	Date	Number of session
	UO:300170	03-03-25	3
	Surname: Álvarez Fernández		
	Name: Andrés		

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo
Escuela de Ingeniería Informática
Universidad de Oviedo

Activity 1. [Divide and Conquer by Subtraction]

SUBSTRACTION1:

This algorithm has one recursive call (a = 1) and in this call n is subtracted one by one (b = 1). The complexity of the algorithm without the recursive call is constant (k = 0), so the final complexity is linear (O(n)).

SUBSTRACTION2:

This algorithm has one recursive call (a = 1) and in this call n is subtracted one by one (b = 1). The complexity of the algorithm without the recursive call is linear (k = 1), so the final complexity is quadratic ($O(n^2)$).

Substraction1 stops giving times after n = 32768 and Substraction2 after n = 16384 because it consumes a lot of memory and causes an overflow in the stack.

SUBSTRACTION3:

This algorithm has two different recursive calls (a = 2) and in both calls n is subtracted one by one (b = 1). The complexity of the algorithm without the recursive calls is constant (k = 0), so the final complexity is exponential ($O(2^n)$).

n	Subtraction4
100	LoR
200	LoR
400	LoR
800	LoR
1600	LoR
3200	209
6400	1461
12800	10804
25600	OoT

n	Subtraction5
30	LoR
32	59
34	60
36	513
38	514
40	4633
42	4567
44	41965
46	41584

<table>
<tr><td rowspan="4">Algorithmics</td><td>Student information</td><td>Date</td><td>Number of session</td></tr>
<tr><td>UO:300170</td><td>03-03-25</td><td>3</td></tr>
<tr><td>Surname: Álvarez Fernández</td><td></td><td></td></tr>
<tr><td>Name: Andrés</td><td></td><td></td></tr>
</table>

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo
Escuela de Ingeniería Informática
Universidad de Oviedo

Activity 2. [Divide and Conquer by Division]

DIVISION1:

This algorithm has one recursive call (a = 1) and in this call n is divided by 3 (b = 3). The complexity of the algorithm without the recursive call is linear (k = 1), so the final complexity is the same as the algorithm without the recursive call (O(n)).

DIVISION2:

This algorithm has two recursive calls (a = 2) and in both calls n is divided by 2 (b = 2). The complexity of the algorithm without the recursive calls is linear (k = 1), so the final complexity is the same as the algorithm times the logarithm of n (O(n*log(n)).

DIVISION3:

This algorithm has two recursive calls (a = 2) and in both calls n is divided by 2 (b = 2). The complexity of the algorithm without the recursive calls is constant (k = 0), so the final complexity is $O(n^{(\log_2(2))})$, which is equal to O(n), that is linear.

n	Division4	Division5
100	LoR	LoR
200	LoR	LoR
400	LoR	LoR
800	LoR	LoR
1600	LoR	53
3200	LoR	497
6400	82	1202
12800	326	12342
25600	1303	29660
51200	5196	OoT
102400	21130	OoT

Activity 3. [Two basic examples]

sum1:

The complexity of this algorithm is linear (O(n)) because it has a loop.

sum2:

This algorithm has one recursive call (a = 1) and in this call n is subtracted one by one (b = 1). The complexity of the algorithm without the recursive call is constant (k = 0), so the final complexity is linear (O(n)).

sum3:

This algorithm has two recursive calls (a = 2) and in both calls n is divided by 2 (b = 2). The complexity of the algorithm without the recursive calls is constant (k = 0), so the final complexity is $O(n^{(\log_2(2))})$, which is equal to O(n), that is linear.

N	Sum1	Sum2	Sum3
3	LoR	LoR	LoR
6	LoR	LoR	LoR
12	LoR	LoR	LoR
24	LoR	LoR	LoR
48	LoR	LoR	81
96	LoR	62	139
192	LoR	121	317
384	84	259	566
768	178	846	1263
1536	363	2158	2216
3072	731	4666	4963
6144	1510	9927	8584
12288	2949	20571	19848
24576	5844	40839	34478
49152	11610	OoT	OoT
98304	23892	OoT	OoT

Algorithmics	Student information	Date	Number of session
	UO:300170	03-03-25	3
	Surname: Álvarez Fernández		
	Name: Andrés		

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo
Escuela de Ingeniería Informática
Universidad de Oviedo

Fib1:

The complexity of this algorithm is linear (O(n)) because it has a loop.

Fib2:

The complexity of this algorithm is linear (O(n)) because it has a loop.

Fib3:

This algorithm has one recursive call (a = 1) and in this call n is subtracted one by one (b = 1). The complexity of the algorithm without the recursive call is constant (k = 0), so the final complexity is linear (O(n)).

Fib4:

This algorithm has two different recursive calls (a = 2). In one of the calls n is subtracted one by one (b = 1) and in the other is subtracted two by two (b = 2). The complexity of the algorithm without the recursive calls is constant (k = 0), so the final complexity in exponential (O(2^n)).

n	Fib1	Fib2	Fib3	Fib4
10	129	135	217	LoR
20	173	306	528	LoR
30	227	451	688	211
40	320	694	845	25102
50	283	926	1129	OoT

Algorithmics	Student information	Date	Number of session
	UO:300170	03-03-25	3
	Surname: Álvarez Fernández		
	Name: Andrés		

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo
Escuela de Ingeniería Informática
Universidad de Oviedo

Activity 4. [Petanque championship organization]

The algorithm shows us all the plays for each day (the number of days is thye number of participants minus one because all participants must play against all the rest participants).

The final algorithm calls the method printParticipantsDay(), which shows the plays that will take place in two days and has a linear complexity and later calls the method moveElementsArray(), which moves the elements of the participants list two positions backwards and has a linear complexity too.

As we call this two methods in write(int index) until the index is equal to the number of participants, the complexity of the final algorithm is quadratic ($O(n^2)$).

n	T Calendar
2	LoR
4	LoR
8	LoR
16	LoR
32	LoR
64	LoR
128	LoR
256	57
512	225
1024	1110
2048	6237
4096	45853
8192	OoT
16384	OoT
32768	OoT

The times obtained in the measurements agree with the teorical complexity, which is $O(n^2)$.