Algorithmics	Student information	Date	Number of session
	UO:269546	19-02-21	1-2
	Surname: Fernández Arias		
	Name: Sara		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. Two algorithms with the same complexity

The following measurements were made using an Intel® Core ™ i7-8550U CPU and a 8 GB RAM. The nTimes used was 1000. Both loops would iterate up to 4096 without crashing.

N	loop2(t)	loop3(t)	loop2(t)/loop3(t)
8	<50ms	<50ms	
16	<50ms	<50ms	
32	<50ms	<50ms	
64	87	51	1,705882353
128	358	178	2,011235955
256	1318	688	1,915697674
512	5525	2742	2,014952589
1024	21942	10960	2,002007299
2048	87419	43933	1,989825416
4096	351781	175611	2,003183172

The first three rows are not used, since the values obtained where smaller than 50ms , and therefore non reliable.

Since both have the same complexity, we'll compare them using the implementation constant. In this case, the division ratio it's greater than 1. Therefore, algorithm of loop3(t) has a better implementation , since it's faster.

Activity 2. Two algorithms with different complexity

The following measurements were made using an Intel® Core ™ i7-8550U CPU and a 8 GB RAM. The nTimes used was 1000. Loop 1 would iterate without crashing up to n=65536 , but , as stated before, loop 2 only reaches 4096 .

N	loop1(t)	loop2(t)	loop1(t)/loop2(t)
8	<50ms	<50ms	
16	<50ms	<50ms	
32	<50ms	<50ms	
64	10	87	0,114942529
128	23	358	0,06424581
256	57	1381	0,041274439
512	111	5525	0,020090498
1024	261	21942	0,011894996
2048	522	87419	0,005971242
4096	1137	351781	0,003232125
8192	2405	-	-
16384	5186	-	-
32768	11144	-	-
65536	23439	-	-

Algorithmics	Student information	Date	Number of session
	UO:269546	19-02-21	1-2
	Surname: Fernández Arias		
	Name: Sara		

The first three rows are not used, since the values obtained where smaller than 50ms , and therefore non reliable.

In this case, the algorithms have different complexities, therefore we'll compare them using their division ratio and evaluating it's tendency. In this case, the division ratio tends to 0, therefore, the numerator which in this case is loop1() it's faster , an therefore , better. This is probably caused because the second loop of algorithm loop1() :

```java
for (int j=1; j<=n; j*=2)
```

The index grows much faster than loop2()'s ones, therefore , it will meet the stopping condition sooner.

```java
for (int j=1; j<=n; j++)
```

Loop1() has a O(n $log\ n$) complexity, while loop 2 has a O(n^2) complexity.

```java
public static void loop1(int n){
    Random rn = new Random();
    @SuppressWarnings("unused")
    int cont = 0;
    for (int i=1; i<=n; i++)
        for (int j=1; j<=n; j*=2)
            cont += rn.nextInt();
}
```

O(n)*O($\log n$)) = O(n log n)

```java
public static void loop2(int n) {
    Random rn = new Random();
    @SuppressWarnings("unused")
    int cont = 0;
    for (int i=1; i<=n; i++)
        for (int j=1; j<=n; j++)
            cont += rn.nextInt();
}
```

O(n)*O(n)= O(n^2)

Activity 3. Complexity of other algorithms

The following measurements were made using an Intel® Core ™ i7-8550U CPU and a 8 GB RAM. The nTimes used was 10.

N	loop4(t)	loop5(t)	loop4(t)/loop5(t)
8	50ms	<50ms	-
16	<50ms	<50ms	-
32	117	106	1,103773585
64	1787	1109	1,611361587
128	28150	9275	3,035040431
256	451210	85805	5,258551366

Algorithmics	Student information	Date	Number of session
	UO:269546	19-02-21	1-2
	Surname: Fernández Arias		
	Name: Sara		

The first two rows are not used, since the values obtained where smaller than 50ms , and therefore non reliable.

We know loop4() has a complexity of O(n^4) while loop5() has a complexity of O(n^3 log n), therefore we will use the tendency of their division ratio in order to evaluate which one is better. In this case, it tends to infinite, therefore, loop5() It's faster, and so , better than loop4(). It makes sense, since exponential complexities are worse the greater their exponents are. In case of loop5(), the exponential part has a smaller exponent, and it's multiplied by a logarithmic complexity , which are known to be one of the fastest.

Activity 4. Study of Unknown.java

The following measurements were made using an Intel® Core ™ i7-8550U CPU and a 8 GB RAM. The nTimes used was 100.

n	unknown
1	<50ms
2	<50ms
4	<50ms
8	<50ms
16	<50ms
32	<50ms
64	72ms
128	437ms
256	2258ms
512	15661ms
1024	118581ms

Escuela de Ingeniería Informática

Algorithmics	Student information	Date	Number of session
	UO:269546	19-02-21	1-2
	Surname: Fernández Arias		
	Name: Sara		

The method has a O(n^3). It complexity takes 72ms(**t1**) for a size of 64(**n1**). The time(**t2**) it will take for a size of 128(**n2**) will be the following:

n2=k*n1 -> 128=k*64 then k= n2/n1 =128/64=2. K =2.

We have that t2=f(n2)/f(n1)* t1 , then for f(n)= O(n^3) ; t2=k^c*t1=2^3*72ms= 576 ms.

The time taken empirically was 437ms. I think it makes sense since there might be other processes of the computer that can be involved in the time , that either slow it or fasten it up.

Now we know that for a size of 128(**n1**), 437ms. (**t1**) were taken. What time(**t2**) should it take for the size of 256? (**n2**)

n2=k*n1 -> 256=k*128 then k= n2/n1 =256/128=2. K =2.

We have that t2=f(n2)/f(n1)* t1 , then for f(n)= O(n^3) ; t2=k^c*t1=2^3*437ms= 3496ms.

The time taken empirically was 2258ms. I think it makes sense since there might be other processes of the computer that can be involved in the time , that either slow it or fasten it up.