

# Trabajo Bioinformática<sup>\*</sup>

Ignacio Amat Hernández

Ángela Llopis Hernández

Estrella Ballester Hoyo

26 de abril de 2020

---

<sup>\*</sup>Grado en Ingeniería Biomédica, Escuela Técnica Superior de Ingenieros Industriales, Valencia, España.

Hemos implementado las trazas de manera que son perfectamente idénticas en Python y en C.

1		Mutation	i	z	k	l
2	Deletion	[l]	1	-1	0	6
3	Insertion	[g]	2	-1	1	2
4	Substitution	[g] $\rightarrow$ [l]	1	-1	1	2
5	Insertion	[l]	2	-1	3	3
6	Match	[l]	1	0	3	3
7	Deletion	[o]	0	-1	3	3
8	Insertion	[o]	1	-1	5	5
9	Match	[o]	0	0	5	5
10	Deletion	[g]	-1	-1	5	5
11	Insertion	[g]	0	-1	1	1
12	Match	[g]	-1	0	1	1
13	<hr/>					
14	Insertion	[o]	2	-1	4	6
15	Substitution	[o] $\rightarrow$ [l]	1	-1	4	6

Cuadro 1: Traza de INEXRECUR con  $X = \text{“googol$”}$ ,  $W = \text{“gol”}$ ,  $z = 0$  en C y Python

[illegible]

Cuadro 2: Traza de INEXRECUR con  $X = \text{"googol\$"}$ ,  $W = \text{"gol"}$ ,  $z = 0$  en R

1		Mutation	i	z	k	l
2	Deletion	[g]	2	-1	0	6
3	Insertion	[g]	3	-1	1	2
4	Match	[g]	2	0	1	2
5	Deletion	[o]	1	-1	1	2
6	Insertion	[o]	2	-1	4	4
7	Match	[o]	1	0	4	4
8	Deletion	[o]	0	-1	4	4
9	Insertion	[o]	1	-1	6	6
10	Match	[o]	0	0	6	6
11	Insertion	[l]	3	-1	3	3
12	Substitution	[l] -> [g]	2	-1	3	3
13	Insertion	[o]	3	-1	4	6
14	Substitution	[o] -> [g]	2	-1	4	6

Cuadro 3: Traza de INEXRECURL con  $X = \text{"googol\$"}$ ,  $W = \text{"goog"}$ ,  $z = 0$  en C y Python

1		INEXRECURL		−	by	XAVI	GABRI	AITANA	ALFREDO		
2	−	D	[ g ]			2	−1	0	6		
3	−	I	[ g ]			3	−1	1	2		
4	−	M	[ g ]			2	0	1	2		
5	−	−	D	[ o ]			1	−1	1	2	
6	−	−	I	[ o ]			2	−1	4	4	
7	−	−	M	[ o ]			1	0	4	4	
8	−	−	−	D	[ o ]			0	−1	4	4
9	−	−	−	I	[ o ]			1	−1	6	6
10	−	−	−	M	[ o ]			0	0	6	6
11	−	I	[ l ]			3	−1	3	3		
12	−	S	[ l −> g ]			2	−1	3	3		
13	−	I	[ o ]			3	−1	4	6		
14	−	S	[ o −> g ]			2	−1	4	6		

Cuadro 4: Traza de INEXRECURL con  $X = \text{"googol\$"}$ ,  $W = \text{"goog"}$ ,  $z = 0$  en R

1		Mutation	i	z	k	l
2	Deletion	[l]	2	0	1	6
3	Deletion	[o]	1	-1	1	6
4	Insertion	[g]	2	-1	1	2
5	Substitution	[g] -> [o]	1	-1	1	2
6	Insertion	[o]	2	-1	4	6
7	Match	[o]	1	0	4	6
8	Deletion	[o]	0	-1	4	6
9	Insertion	[g]	1	-1	1	2
10	Substitution	[g] -> [o]	0	-1	1	2
11	Insertion	[o]	1	-1	6	6
12	Match	[o]	0	0	6	6
13	Deletion	[g]	-1	-1	6	6
14	Insertion	[g]	0	-1	2	2
15	Match	[g]	-1	0	2	2
16	<hr style="border-top: 1px solid black;"/> {2,2} <hr style="border-top: 1px solid black;"/>					
17	Insertion	[g]	3	0	1	2
18	Substitution	[g] -> [l]	2	0	1	2
19	Deletion	[o]	1	-1	1	2
20	Insertion	[o]	2	-1	4	4
21	Match	[o]	1	0	4	4
22	Deletion	[o]	0	-1	4	4
23	Insertion	[o]	1	-1	6	6
24	Match	[o]	0	0	6	6
25	Deletion	[g]	-1	-1	6	6
26	Insertion	[g]	0	-1	2	2
27	Match	[g]	-1	0	2	2
28	<hr style="border-top: 1px solid black;"/> {2,2} <hr style="border-top: 1px solid black;"/>					
29	Insertion	[o]	3	0	4	6
30	Substitution	[o] -> [l]	2	0	4	6
31	Deletion	[o]	1	-1	4	6
32	Insertion	[g]	2	-1	1	2
33	Substitution	[g] -> [o]	1	-1	1	2
34	Insertion	[o]	2	-1	6	6
35	Match	[o]	1	0	6	6
36	Deletion	[o]	0	-1	6	6
37	Insertion	[g]	1	-1	2	2
38	Substitution	[g] -> [o]	0	-1	2	2

Cuadro 5: Traza de INEXRECUR con X = “googol\$”, W = “gool”, z = 1 en C y Python

[illegible]

Cuadro 6: Traza de INEXRECUR con  $X = \text{"googol\$"}, W = \text{"gool"}, z = 1$  en R

```

1  inexrecur_time.c
2  12.34  $\mu$ s from 10000 iterations.
3
4  inexrecur_time.py
5  real      sys      user
6  268.89 $\mu$ s  0.37 $\mu$ s  268.16 $\mu$ s
7
8  inexrecur_time.R
9  user      system    elapsed
10 5422 $\mu$ s    18 $\mu$ s      5443 $\mu$ s

```

Cuadro 7: Tiempos de “CPU”.

Ejecutando desde la línea de comandos como scripts usando `#!/bin/env Rscript` y `#!/bin/env Python`.

```

1  bench ./inexrecur_clean ./inexrecur_clean.py ./inexrecur_clean.R
2  benchmarking bench/./inexrecur_clean
3  time                4.524 ms      (4.496 ms .. 4.551 ms)
4                      0.999 R2      (0.999 R2 .. 1.000 R2)
5  mean                4.522 ms      (4.499 ms .. 4.559 ms)
6  std dev             89.83  $\mu$ s      (58.07  $\mu$ s .. 133.1  $\mu$ s)
7
8  benchmarking bench/./inexrecur_clean.py
9  time                39.26 ms      (39.12 ms .. 39.40 ms)
10                      1.000 R2      (1.000 R2 .. 1.000 R2)
11  mean                39.30 ms      (39.18 ms .. 39.45 ms)
12  std dev             266.5  $\mu$ s      (177.4  $\mu$ s .. 388.7  $\mu$ s)
13
14  benchmarking bench/./inexrecur_clean.R
15  time                278.5 ms      (273.6 ms .. 285.0 ms)
16                      1.000 R2      (1.000 R2 .. 1.000 R2)
17  mean                280.6 ms      (279.1 ms .. 281.9 ms)
18  std dev             1.714 ms      (1.027 ms .. 2.517 ms)
19  variance introduced by outliers: 16% (moderately inflated)

```

Cuadro 8: Tiempo de “pared” según la utilidad *bench*.

```

1  inexrecur_clean.c
2  ==81469==      in use at exit: 18,588 bytes in 164 blocks
3  ==81469==      total heap usage: 191 allocs , 27 frees , 24,894 bytes allocated
4
5  inexrecur_mem.py
6  6,631,424 bytes
7
8  inexrecur_mem.R
9  4,932,584 bytes

```

Cuadro 9: Memoria usada medido usando herramientas de los lenguajes.

```

1  time ./inexrecur_clean
2  (5,5)
3  (1,1)
4  ./inexrecur_clean
5  0.00s  user 0.00s system 44% cpu 0.004 total
6  max memory:          676 kB
7
8  time ./inexrecur_clean.py
9  (1, 1)
10 (5, 5)
11 ./inexrecur_clean.py
12 0.03s  user 0.01s system 82% cpu 0.039 total
13 max memory:          6272 kB
14
15 time ./inexrecur_clean.R
16 [[1]]
17 [1] 5 5
18
19 [[2]]
20 [1] 1 1
21
22 ./inexrecur_clean.R
23 0.23s  user 0.04s system 97% cpu 0.277 total
24 max memory:          67476 kB

```

Cuadro 10: Memoria usada medido usando herramientas del sistema operativo.