

DEMOSTRACIÓN DEL PRINCIPIO DE INVARIANZA CON LOS PROGRAMAS “BurbujaC.cpp” Y “BurbujaJava.java”

The screenshot displays the NetBeans IDE interface. The left pane, titled 'Output - JavaApplication1 (run)', shows the execution results of 'BurbujaJava.java'. The right pane, titled 'Terminal', shows the execution results of 'BurbujaC.cpp'.

Left Pane Output (JavaApplication1):

```
run:
N: 1000 T (ms.): 9
N: 2000 T (ms.): 12
N: 3000 T (ms.): 8
N: 4000 T (ms.): 13
N: 5000 T (ms.): 21
N: 6000 T (ms.): 30
N: 7000 T (ms.): 41
N: 8000 T (ms.): 50
N: 9000 T (ms.): 62
N: 10000 T (ms.): 81
N: 11000 T (ms.): 92
N: 12000 T (ms.): 107
N: 13000 T (ms.): 127
N: 14000 T (ms.): 153
N: 15000 T (ms.): 173
N: 16000 T (ms.): 198
N: 17000 T (ms.): 220
N: 18000 T (ms.): 245
N: 19000 T (ms.): 273
N: 20000 T (ms.): 304
BUILD SUCCESSFUL (total time: 2 seconds)
```

Right Pane Output (Terminal):

```
a.out r9.@jjavier98-Lenovo-ideapad-310-15IKB:~/GII/2º/2º Semestre/ALG
N: 1000 T (ms.): 3.034
N: 2000 T (ms.): 11.172
N: 3000 T (ms.): 25.808
N: 4000 T (ms.): 43.068
N: 5000 T (ms.): 68.551
N: 6000 T (ms.): 97.226
N: 7000 T (ms.): 131.61
N: 8000 T (ms.): 171.186
N: 9000 T (ms.): 216.78
N: 10000 T (ms.): 266.186
N: 11000 T (ms.): 321.225
N: 12000 T (ms.): 380.729
N: 13000 T (ms.): 448.608
N: 14000 T (ms.): 519.968
N: 15000 T (ms.): 599.022
N: 16000 T (ms.): 680.835
N: 17000 T (ms.): 767.994
N: 18000 T (ms.): 862.373
N: 19000 T (ms.): 958.53
N: 20000 T (ms.): 1062.01
m$
```

A la izquierda 'BurbujaJava.java' ejecutado en Netbeans
A la derecha 'BurbujaC.cpp' compilado sin optimización (g++ BurbujaC.cpp)

El principio de invarianza nos indica que dos implementaciones de un mismo algoritmo difieren en su tiempo de ejecución en una constante multiplicativa de manera que, siendo el tamaño del problema 'x' grande, si $t_1(x)$ y $t_2(x)$ son los tiempos de ejecución en las distintas implementaciones, podemos afirmar que:

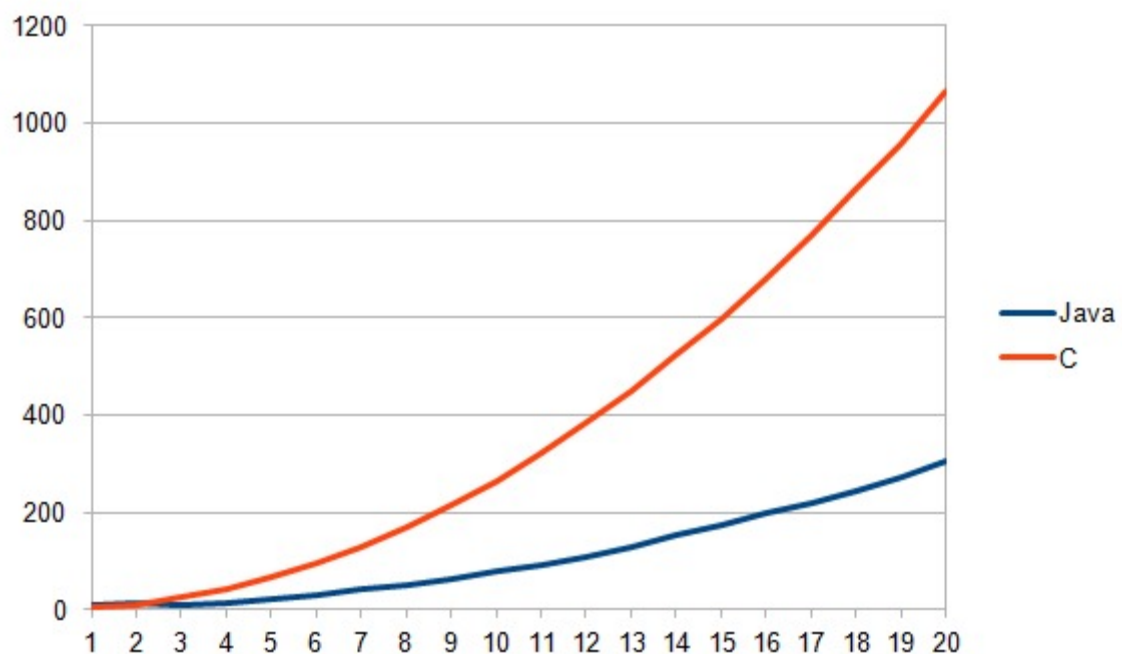
$$t_1(x) = c \cdot t_2(x) \quad || \quad c \cdot t_1(x) = t_2(x)$$

Siendo 'c' positiva y dependiendo, junto con los demás valores (excepto 'x'), de la computadora.

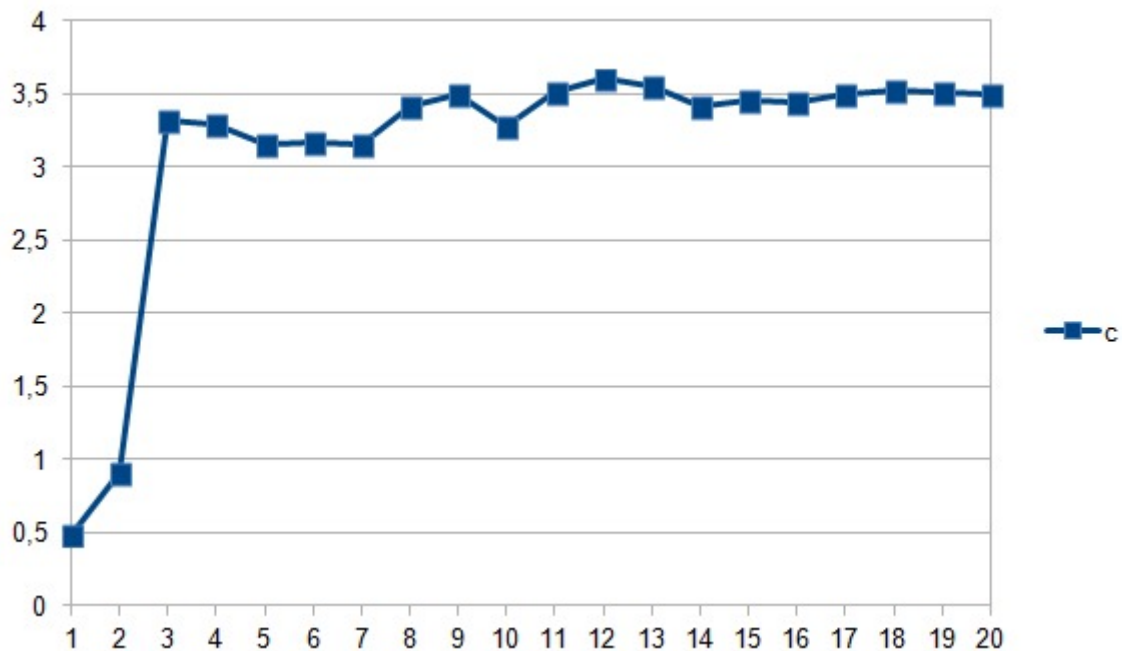
Para hallar 'c' despejamos:

$$c = t_2(X) / t_1(X)$$

Con los programas aportados y los datos obtenidos obtenemos los siguientes diagramas:



Como se observa, en lo que respecta a tamaños de problema pequeños, no importa la implementación del algoritmo que escojamos. Sin embargo a medida que aumentamos el tamaño la diferencia de tiempos se hace mucho más considerable.



En lo que respecta a la constante multiplicativa 'c', como hemos indicado en la definición del principio de invarianza, se hace constante con tamaños de problema grandes tal y como se ve en la gráfica. La 'c' se va unificando en torno al 3,5 a medida que aumenta el tamaño.