1.8

Vantablack

2025-04-21

La penitencia de Newton

Comentan algunos historiadores que Newton tenía en le escuela primaria un profesor muy exigente que reclamaba constantemente la atención de los alumnos a lo que el estaba enseñando. Tan pronto alguien cometía un acto improcedente lo castigaba con una penitencia que consistía en traer para el día siguiente el resultado de la suma de desde el número 1 hasta cierto número aleatoriamente elegido. Por supuesto esta tarea demandaría quedarse toda la noche haciendo las sumas parciales, tarea que el profesor ya había realizado previamente y que tenía registrado en un voluminoso cuaderno que (al esltio tabla de logaritmos) cargaba religiosamente en su portafolios de cuero.

Consigna de trabajo : Desarrolla dos algoritmos que hagan este trabajo , por ejemplo para sumar desde 1 hasta 1.1010 y verifica cuál de los dos es más eficiente.

```
# Método 1: Usando la fórmula matemática de Gauss
# La suma de números desde 1 hasta n es n*(n+1)/2
suma_formula <- function(n) {
  return(n * (n + 1) / 2)
}

# Método 2: Iteración (suma uno por uno)
suma_iteracion <- function(n) {
  suma <- 0
  for (i in 1:n) {
    suma <- suma + i
  }
  return(suma)
}

# Comparación de eficiencia para números pequeños
library(microbenchmark)

# Comparamos con n = 10^5 (un número más pequeño para que la iteración sea factible)
n_pequeno <- 10^5
resultado_pequeno <- microbenchmark(
  formula = suma_formula(n_pequeno),
  iteracion = suma_iteracion(n_pequeno),
  times = 100
)
print(resultado_pequeno)
```

```
## Unit: nanoseconds
##       expr     min      lq       mean  median      uq     max neval
##    formula     280     380   20091.83     675    1170 1914362   100
##  iteracion 1366851 1408152 1477737.67 1433981 1472476 3457733   100
```

```r
# Para n = 10^10, sólo es práctico usar la fórmula
n_grande <- 10^10
cat("\nSuma de 1 hasta 10^10 usando la fórmula:\n")
```

```
##
## Suma de 1 hasta 10^10 usando la fórmula:
```

```r
tiempo_inicio <- Sys.time()
resultado_grande <- suma_formula(n_grande)
tiempo_fin <- Sys.time()
cat("Resultado:", resultado_grande, "\n")
```

```
## Resultado: 5e+19
```

```r
cat("Tiempo:", difftime(tiempo_fin, tiempo_inicio, units = "secs"), "segundos\n")
```

```
## Tiempo: 0.0006871223 segundos
```

```r
# Intentar calcular con iteración para n = 10^10 sería extremadamente lento
# Esto lo podemos demostrar con un n más pequeño para extrapolar
n_medio <- 10^7
cat("\nMidiendo el tiempo para n = 10^7 con iteración (para extrapolar):\n")
```

```
##
## Midiendo el tiempo para n = 10^7 con iteración (para extrapolar):
```

```r
tiempo_inicio <- Sys.time()
resultado_medio <- suma_iteracion(n_medio)
tiempo_fin <- Sys.time()
tiempo_iteracion <- difftime(tiempo_fin, tiempo_inicio, units = "secs")
cat("Tiempo para 10^7:", tiempo_iteracion, "segundos\n")
```

```
## Tiempo para 10^7: 0.1329508 segundos
```

```r
# Extrapolación para n = 10^10
tiempo_estimado <- as.numeric(tiempo_iteracion) * 1000 # 10^10 es 1000 veces más grande que 10^7
cat("\nTiempo estimado para n = 10^10 con iteración:", tiempo_estimado/60/60, "horas\n")
```

```
##
## Tiempo estimado para n = 10^10 con iteración: 0.03693077 horas
```

Este código compara ambos métodos y muestra claramente por qué la fórmula matemática es mucho más eficiente. Para la suma hasta 10^10:

Método de la fórmula (Gauss): Calcula instantáneamente usando la fórmula n(n+1)/2 Método iterativo: Requeriría un tiempo extremadamente largo (como se muestra en la extrapolación)

La moraleja de esta historia (y posiblemente lo que Newton descubrió) es que las matemáticas pueden proporcionar soluciones elegantes que evitan cálculos repetitivos y tediosos. La fórmula n(n+1)/2 resuelve instantáneamente lo que de otra manera sería una tarea imposible de hacer manualmente.

Algunos aspectos importantes:

El método iterativo tiene complejidad O(n), lo que significa que el tiempo crece linealmente con n.

El método de la fórmula tiene complejidad O(1), es decir, tiempo constante independientemente de n.

Para n = 10^10, la solución iterativa sería inviable mientras que la fórmula da el resultado al instante.

La suma de los números del 1 al 10^10 es 5×10^19 (50,000,000,000,000,000,000), un número extremadamente grande que habría sido imposible de calcular manualmente en la época de Newton.