

Atividade: Tempo de versões do Fibonacci

Nome: Gabriel Henrique Silva Duque R.A: 0082574

Código Fonte R:

```

rm(list = ls())
cat("\014")

fibonacciIterativo <- function(n) {
  if (n <= 2) return(1)
  vetor <- numeric(n)
  vetor[1] <- 1
  vetor[2] <- 1
  for (i in 3:n) {
    vetor[i] <- vetor[i - 1] + vetor[i - 2]
  }
  return(vetor)
}
print(fibonacciIterativo(26))

#-----

fibonacciRecursivo <- function(n) {
  if (n <= 2) {
    return(1)
  } else return(fibonacciRecursivo(n - 1) + fibonacciRecursivo(n - 2))
}
print(fibonacciRecursivo(25))

#-----

nexec <- 30
inicio <- 2
fim <- 25
n <- fim - inicio + 1
tamanhos <- inicio:fim

tempos_fibonacciIterativo <- rep (0, n)
tempos_fibonacciRecursivo <- rep (0, n)

# Coleta dos tempos de execução
for (i in tamanhos) {
  idx <- i - inicio + 1
  tempos_iterativo <- numeric(nexec)
  tempos_recursivo <- numeric(nexec)

  for (j in 1:nexec) {
    tempos_iterativo[j] <- system.time(fibonacciIterativo(i))["elapsed"]
    tempos_recursivo[j] <- system.time(fibonacciRecursivo(i))["elapsed"]
  }
}

```

```

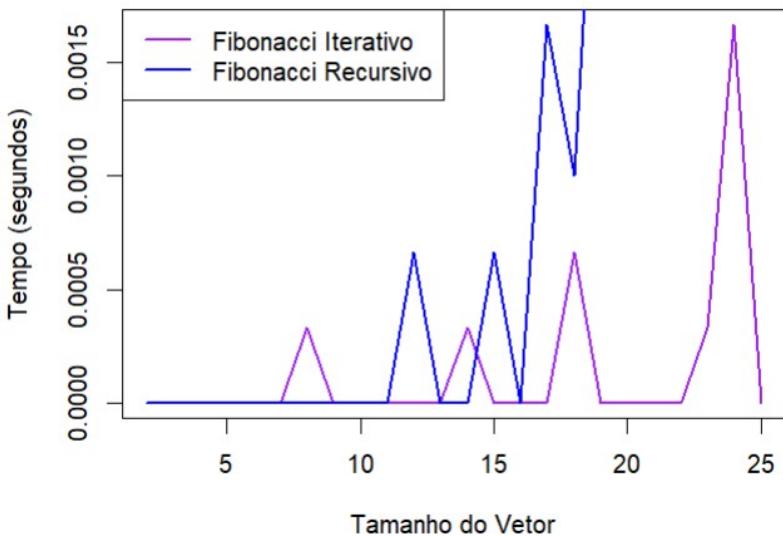
#armazenamos os tempos num vetor
tempos_fibonacciIterativo[idx] <- mean(tempos_iterativo)
tempos_fibonacciRecursivo[idx] <- mean(tempos_recursivo)
}

plot(
  tamanhos,
  tempos_fibonacciIterativo,
  type = "l",
  col = "purple",
  lwd = 2,
  xlab = "Tamanho do Vetor",
  ylab = "Tempo (segundos)",
  main = "Desempenho: Fibonacci Iterativo x Fibonacci Recursivo"
)
lines(tamanhos,
      tempos_fibonacciRecursivo,
      col = "blue",
      lwd = 2)
legend(
  "topleft",
  legend = c("Fibonacci Iterativo", "Fibonacci Recursivo"),
  col = c("purple", "blue"),
  lwd = 2
)

```

Imagen gerada no Rstudio:

Desempenho: Fibonacci Iterativo x Fibonacci Recursivo



Respondendo a questão 5:

A versão iterativa apresenta complexidade $O(n)$ e seu tempo de execução cresce de forma proporcional a n como podemos observar no gráfico.

Já a versão recursiva possui complexidade exponencial $O(2^n)$, e isso é refletido no gráfico com crescimento acelerado para valores muito grandes de n .