Relatório - Prática 2

1. Plano de experimentos:

O objetivo dessa prática é avaliar o comportamento de funções de acordo com diferentes abordagens(Iterativa e Recursiva). O intervalo de testes abrangeu entradas de 1 a 20 para ambas as funções. Esse intervalo foi escolhido devido à limitação do tipo de dado "long long int" na linguagem C, que não suporta valores maiores, resultando em saídas incorretas.

2. Resultado dos testes e a relação com a complexidade dos algoritmos:

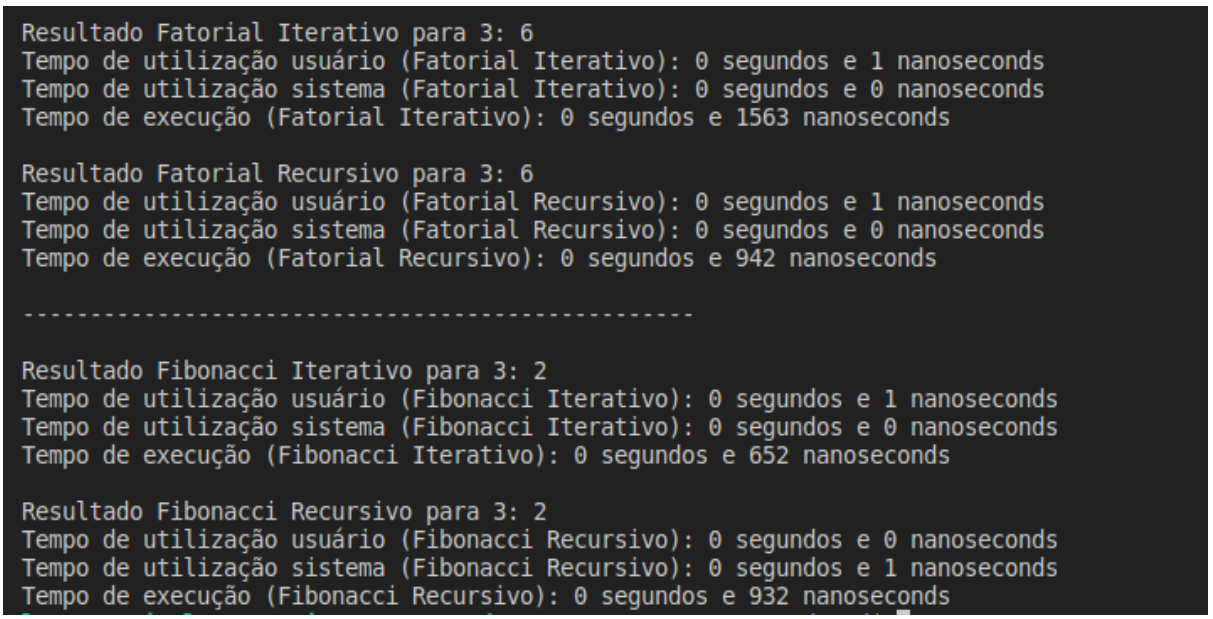
Resultado para a entrada 1:

```
Resultado Fatorial Iterativo para 1: 1
Tempo de utilização usuário (Fatorial Iterativo): 0 segundos e 0 nanoseconds
Tempo de utilização sistema (Fatorial Iterativo): 0 segundos e 1 nanoseconds
Tempo de execução (Fatorial Iterativo): 0 segundos e 1834 nanoseconds

Resultado Fatorial Recursivo para 1: 1
Tempo de utilização usuário (Fatorial Recursivo): 0 segundos e 0 nanoseconds
Tempo de utilização sistema (Fatorial Recursivo): 0 segundos e 3053 nanoseconds
Tempo de execução (Fatorial Recursivo): 0 segundos e 3052290 nanoseconds

------------------------------------------------

Resultado Fibonacci Iterativo para 1: 1
Tempo de utilização usuário (Fibonacci Iterativo): 0 segundos e 0 nanoseconds
Tempo de utilização sistema (Fibonacci Iterativo): 0 segundos e 0 nanoseconds
Tempo de execução (Fibonacci Iterativo): 0 segundos e 782 nanoseconds

Resultado Fibonacci Recursivo para 1: 1
Tempo de utilização usuário (Fibonacci Recursivo): 0 segundos e 0 nanoseconds
Tempo de utilização sistema (Fibonacci Recursivo): 0 segundos e 0 nanoseconds
Tempo de execução (Fibonacci Recursivo): 0 segundos e 682 nanoseconds
```

Resultado para a entrada 2:

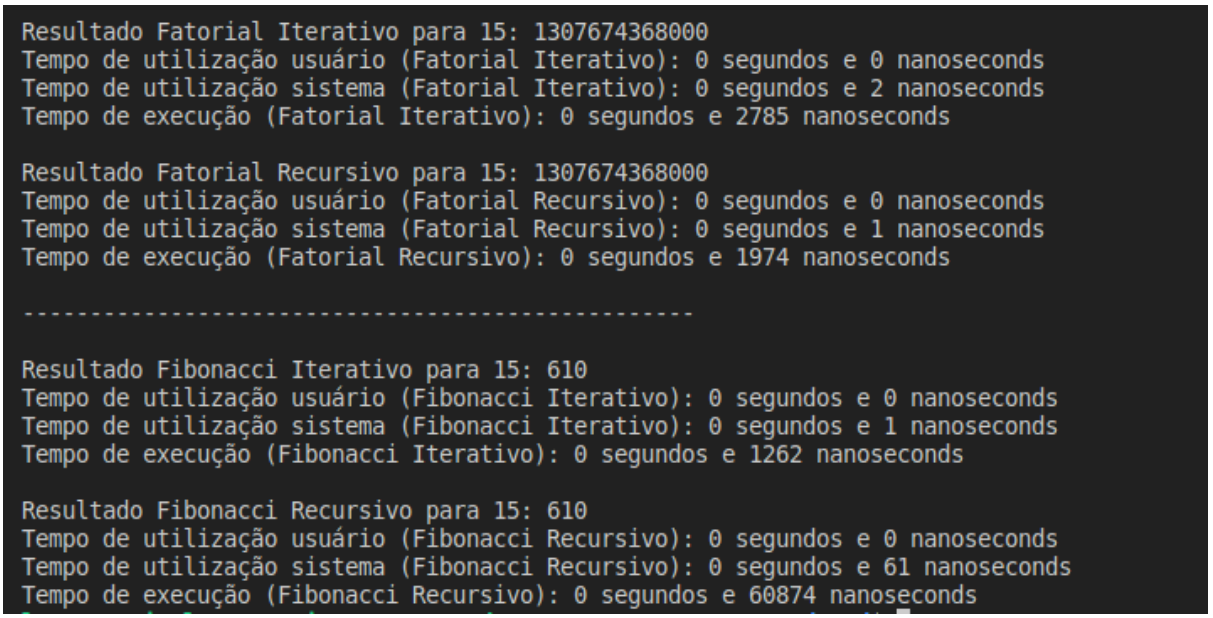
```
Resultado Fatorial Iterativo para 2: 2
Tempo de utilização usuário (Fatorial Iterativo): 0 segundos e 0 nanoseconds
Tempo de utilização sistema (Fatorial Iterativo): 0 segundos e 1 nanoseconds
Tempo de execução (Fatorial Iterativo): 0 segundos e 2615 nanoseconds

Resultado Fatorial Recursivo para 2: 2
Tempo de utilização usuário (Fatorial Recursivo): 0 segundos e 0 nanoseconds
Tempo de utilização sistema (Fatorial Recursivo): 0 segundos e 2 nanoseconds
Tempo de execução (Fatorial Recursivo): 0 segundos e 1562 nanoseconds

------------------------------------------------

Resultado Fibonacci Iterativo para 2: 1
Tempo de utilização usuário (Fibonacci Iterativo): 0 segundos e 0 nanoseconds
Tempo de utilização sistema (Fibonacci Iterativo): 0 segundos e 1 nanoseconds
Tempo de execução (Fibonacci Iterativo): 0 segundos e 1172 nanoseconds

Resultado Fibonacci Recursivo para 2: 1
Tempo de utilização usuário (Fibonacci Recursivo): 0 segundos e 0 nanoseconds
Tempo de utilização sistema (Fibonacci Recursivo): 0 segundos e 1 nanoseconds
Tempo de execução (Fibonacci Recursivo): 0 segundos e 1332 nanoseconds
```

Resultado para a entrada 3:

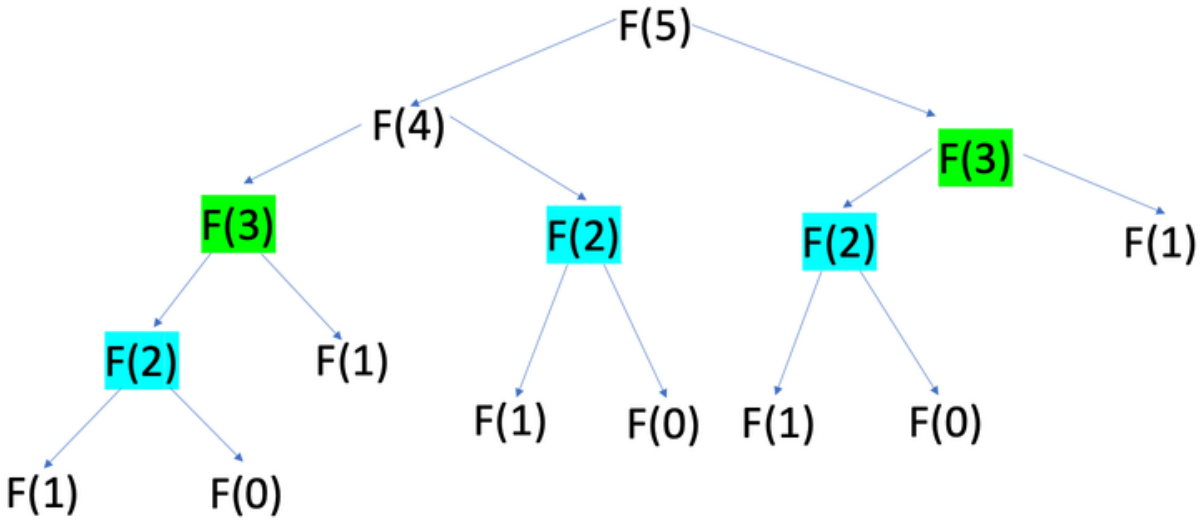


Para as entradas 1, 2 e 3, os resultados apresentaram diferenças mínimas no tempo de execução entre as funções iterativas e recursivas. Entretanto, já é possível observar que o tempo de execução do cálculo do fatorial permanece relativamente constante. Isso pois a complexidade das duas abordagens é igual, O(n).

Vejamos como fica para a entrada 15:



A diferença entre as implementações iterativa e recursiva da sequência de Fibonacci tornou-se mais evidente com a entrada 15. Isso ocorre porque a versão recursiva da função chama repetidamente os valores de n - 1 e n - 2, resultando em um aumento exponencial no número de chamadas recursivas.



Por fim, para entrada 20:

```
Resultado Fatorial Iterativo para 20: 2432902008176640000
Tempo de utilização usuário (Fatorial Iterativo): 0 segundos e 1 nanoseconds
Tempo de utilização sistema (Fatorial Iterativo): 0 segundos e 0 nanoseconds
Tempo de execução (Fatorial Iterativo): 0 segundos e 1614 nanoseconds

Resultado Fatorial Recursivo para 20: 2432902008176640000
Tempo de utilização usuário (Fatorial Recursivo): 0 segundos e 1 nanoseconds
Tempo de utilização sistema (Fatorial Recursivo): 0 segundos e 0 nanoseconds
Tempo de execução (Fatorial Recursivo): 0 segundos e 1323 nanoseconds

------------------------------------------------

Resultado Fibonacci Iterativo para 20: 6765
Tempo de utilização usuário (Fibonacci Iterativo): 0 segundos e 1 nanoseconds
Tempo de utilização sistema (Fibonacci Iterativo): 0 segundos e 0 nanoseconds
Tempo de execução (Fibonacci Iterativo): 0 segundos e 1142 nanoseconds

Resultado Fibonacci Recursivo para 20: 6765
Tempo de utilização usuário (Fibonacci Recursivo): 0 segundos e 305 nanoseconds
Tempo de utilização sistema (Fibonacci Recursivo): 0 segundos e 0 nanoseconds
Tempo de execução (Fibonacci Recursivo): 0 segundos e 303918 nanoseconds
```

Quando testamos para a entrada 20, a diferença entre as abordagens se torna ainda mais notável.

3. Uso do gprof para depuração de desempenho

Utilizamos a ferramenta gprof para obter um perfil detalhado da execução do algoritmo recursivo. Contudo, os resultados não revelaram diferenças significativas para valores de entrada pequenos devido à rápida execução das funções e à granularidade da amostragem de tempo.

No entanto, ao introduzir uma função que consome considerável poder computacional nas chamadas recursivas, observamos uma mudança no perfil de execução. A função ComputeSin, que realiza chamadas recursivas para calcular o seno um milhão de vezes.

Temos esse perfil raso gerado pelo gprof:

```
Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
100.00     47.78    47.78    10965     0.00     0.00  computeSin
  0.00     47.78     0.00        1     0.00     0.00  FatItr
  0.00     47.78     0.00        1     0.00     0.09  FatRec
  0.00     47.78     0.00        1     0.00     0.00  FibItr
  0.00     47.78     0.00        1     0.00    47.69  FibRec
```

E para as funções recursivas temos esse Call graph :

```
index % time    self  children    called     name
                0.09    0.00      20/10965       FatRec [4]
               47.69    0.00   10945/10965       FibRec [3]
[1]    100.0   47.78    0.00   10965          computeSin [1]
```

A função ComputeSin que realiza o cálculo do seno foi a função que consumiu a maior parte do tempo de execução, representando 100% do tempo total de CPU, com um tempo acumulado de 47.78 segundos. Isso se deve ao alto custo computacional dessa função.

A função FatRec (Fatorial Recursivo) agora mostra um tempo de execução significativamente maior em comparação com o experimento anterior. O tempo de execução da função FatRec é de 0.09 segundos, que é principalmente devido à

chamada recursiva para calcular o fatorial. No entanto, em termos de tempo total de CPU, ainda é relativamente baixo.

A função FibRec (Fibonacci Recursivo) também experimentou um aumento considerável no tempo de execução, agora totalizando 47.69 segundos. Isso ocorreu porque, durante a execução recursiva, a função ComputeSin foi chamada repetidamente, consumindo uma quantidade significativa de recursos computacionais.

As funções FatItr (Fatorial Iterativo) e FibItr (Fibonacci Iterativo) não foram afetadas pela introdução da chamada recursiva com consumo de recursos, mantendo um tempo de execução baixo.