RELATÓRIO ATIVIDADE PRÁTICA

A atividade tem como objetivo analisar o tempo de relógio e de usuário dos algoritmos para cálculo do fatorial ou fibonacci de um determinado número. De acordo com as informações dadas pelo gprof e as funções clock_gettime e getrusage, é possível notar uma diferença entre os os algoritmos e correlacionar essa diferença com a complexidade de cada um deles.

Compilacao

Para compilar basta:
> cd <diretoriodestino>
> make run

Com os comandos acima, o programa executará cada versão de cada algoritmo com o parâmetro 5 e apenas uma vez.

Teste unitário

São providos quatro testes: fatorial iterativo e recursivo / fibonacci iterativo e recursivo.
O programa pode ser testado com os seguintes comandos:
> cd <diretoriodestino>
> make test
> bin/main <tipo> <vezes_execucao>

Onde <tipo> pode ser:
- fati (fatorial iterativo)
- fatr (fatorial recursivo)
- fibi (fibonacci iterativo)
- fibr (fibonacci recursivo)

E <vezes_execucao> se refere a quantidade de vezes que a operação escolhida será executada.
Por exemplo:
> bin/main fatr 50
O comando acima executará a funcao de fatorial recursivo 50 vezes.

Plano de experimentos

Para a bateria de testes, foram escolhidos os números 10 e 20. Além disso, os algoritmos foram executados 100 mil e 10 milhões de vezes, respectivamente.

Complexidade dos algoritmos para efeito de comparação:

- Fatorial iterativo: O(n)
- Fatorial recursivo: O(n)
- Fibonacci iterativo: O(n)
- Fibonacci recursivo: O(2^n)

Número 10:

```
bin/main fati 10 100000 /tmp/analisefatigprof.out
Answer: 3628800
Clock time: 0.002010
CPU time: 0.001792 sec user, 0.000000 sec system
gprof bin/main gmon.out > /tmp/analisefatigprof.txt

bin/main fatr 10 100000 /tmp/analisefatrgprof.out
Answer: 3628800
Clock time: 0.011067
CPU time: 0.010368 sec user, 0.000076 sec system
gprof bin/main gmon.out > /tmp/analisefatrgprof.txt

bin/main fibi 10 100000 /tmp/analisefibigprof.out
Answer: 55
Clock time: 0.002521
CPU time: 0.002437 sec user, 0.000000 sec system
gprof bin/main gmon.out > /tmp/analisefibigprof.txt

bin/main fibr 10 100000 /tmp/analisefibrgprof.out
Answer: 55
Clock time: 0.118003
CPU time: 0.114658 sec user, 0.000000 sec system
gprof bin/main gmon.out > /tmp/analisefibrgprof.txt
```

```
bin/main fati 10 10000000 /tmp/analisefatigprof.out
Answer: 3628800
Clock time: 0.178111
CPU time: 0.176045 sec user, 0.000052 sec system
gprof bin/main gmon.out > /tmp/analisefatigprof.txt

bin/main fatr 10 10000000 /tmp/analisefatrgprof.out
Answer: 3628800
Clock time: 1.082814
CPU time: 1.049674 sec user, 0.000989 sec system
gprof bin/main gmon.out > /tmp/analisefatrgprof.txt

bin/main fibi 10 10000000 /tmp/analisefibigprof.out
Answer: 55
Clock time: 0.244612
CPU time: 0.243173 sec user, 0.000087 sec system
gprof bin/main gmon.out > /tmp/analisefibigprof.txt

bin/main fibr 10 10000000 /tmp/analisefibrgprof.out
Answer: 55
Clock time: 11.509402
CPU time: 11.445769 sec user, 0.002902 sec system
gprof bin/main gmon.out > /tmp/analisefibrgprof.txt
```

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  ns/call  ns/call  name
 80.07      0.12     0.12 10000001    12.01    12.01  retornaFatorialRecursivo(int)
 20.02      0.15     0.03                             retornaFatorialIterativo(int)
  0.00      0.15     0.00        7     0.00     0.00  std::__is_constant_evaluated()
  0.00      0.15     0.00        6     0.00     0.00  std::operator&(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00      0.15     0.00        5     0.00     0.00  std::char_traits<char>::length(char const*)
  0.00      0.15     0.00        3     0.00     0.00  std::__new_allocator<char>::~__new_allocator()
  0.00      0.15     0.00        3     0.00     0.00  void std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*
  0.00      0.15     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::basic_string<std::allocator<cl
  0.00      0.15     0.00        3     0.00     0.00  std::ios_base::setf(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00      0.15     0.00        3     0.00     0.00  std::setprecision(int)
  0.00      0.15     0.00        3     0.00     0.00  std::operator&=(std::_Ios_Fmtflags&, std::_Ios_Fmtflags)
  0.00      0.15     0.00        3     0.00     0.00  std::operator~(std::_Ios_Fmtflags)
  0.00      0.15     0.00        3     0.00     0.00  std::operator|=(std::_Ios_Fmtflags&, std::_Ios_Fmtflags)
  0.00      0.15     0.00        3     0.00     0.00  std::operator|(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00      0.15     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*>(cha
  0.00      0.15     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*>(cha
  0.00      0.15     0.00        2     0.00     0.00  int __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsi
  0.00      0.15     0.00        2     0.00     0.00  std::char_traits<char>::compare(char const*, char const*, unsigned long)
  0.00      0.15     0.00        2     0.00     0.00  std::__cxx11::stoi(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, uns
  0.00      0.15     0.00        2     0.00     0.00  bool std::operator==<char, std::char_traits<char>, std::allocator<char> >(std::__cxx11::basic_string<char, std
  0.00      0.15     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00      0.15     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00      0.15     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00      0.15     0.00        1     0.00     0.00  diffUserTime(rusage*, rusage*)
  0.00      0.15     0.00        1     0.00     0.00  diffSystemTime(rusage*, rusage*)
```

```
Flat profile:                                                                    /tmp/analisefibrgprof.txt

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  ns/call  ns/call  name
 92.06      1.49     1.49 10000001   149.13   149.13  retornaFibonacciRecursivo(int)
  7.41      1.61     0.12                             retornaFibonacciIterativo(int)
  0.62      1.62     0.01                             main
  0.00      1.62     0.00       11     0.00     0.00  std::__is_constant_evaluated()
  0.00      1.62     0.00        7     0.00     0.00  std::char_traits<char>::length(char const*)
  0.00      1.62     0.00        6     0.00     0.00  std::operator&(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00      1.62     0.00        4     0.00     0.00  std::char_traits<char>::compare(char const*, char const*, unsigned long)
  0.00      1.62     0.00        4     0.00     0.00  bool std::operator==<char, std::char_traits<char>, std::allocator<char> >(std::__cxx11::basic_string<char, std
  0.00      1.62     0.00        3     0.00     0.00  std::__new_allocator<char>::~__new_allocator()
  0.00      1.62     0.00        3     0.00     0.00  void std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*
  0.00      1.62     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::basic_string<std::allocator<cl
  0.00      1.62     0.00        3     0.00     0.00  std::ios_base::setf(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00      1.62     0.00        3     0.00     0.00  std::setprecision(int)
  0.00      1.62     0.00        3     0.00     0.00  std::operator&=(std::_Ios_Fmtflags&, std::_Ios_Fmtflags)
  0.00      1.62     0.00        3     0.00     0.00  std::operator~(std::_Ios_Fmtflags)
  0.00      1.62     0.00        3     0.00     0.00  std::operator|=(std::_Ios_Fmtflags&, std::_Ios_Fmtflags)
  0.00      1.62     0.00        3     0.00     0.00  std::operator|(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00      1.62     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*>(cha
  0.00      1.62     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*>(cha
  0.00      1.62     0.00        2     0.00     0.00  int __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsi
  0.00      1.62     0.00        2     0.00     0.00  std::__cxx11::stoi(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, uns
  0.00      1.62     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00      1.62     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00      1.62     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00      1.62     0.00        1     0.00     0.00  diffUserTime(rusage*, rusage*)
  0.00      1.62     0.00        1     0.00     0.00  diffSystemTime(rusage*, rusage*)
```

Com a repetição de 10 milhões de vezes dos algoritmos fica mais claro a diferença entre a implementação iterativa e recursiva de cada um, principalmente a implementação recursiva de fibonacci. Essa disparidade de tempo para os algoritmos recursivos se dá pela razão de que a recursividade utilizada dessa forma não aproveita valores que já foram calculados. Por exemplo, no algoritmo de fibonacci, se feita uma árvore com os valores calculados, quanto maior é o valor de entrada, mais os valores se repetem já que a árvore continua crescendo para baixo e usa todos os seus vértices sem nenhum tipo de otimização, diferentemente do algoritmo iterativo.

Dadas as complexidades citadas acima, é completamente normal ver essa discrepância do algoritmo de fibonacci recursivo para o restante, haja vista que possui complexidade exponencial, muito maior do que O(n). Apesar de também haver uma diferença no tempo do fatorial iterativo e recursivo, não é tão grande como na comparação anterior.

Com relação ao tempo de relógio e tempos de usuário / sistema, é possível ver uma grande semelhança nos tempos de relógio e de usuário em todos os casos de teste. Em contrapartida, o tempo de sistema sempre esteve próximo do 0.

É possível notar nos prints do gprof a quantidade de vezes que a função foi chamada e que há uma leve discrepância com o tempo de relógio printado no programa. O tempo total estipulado no teste de 100 mil execuções do fatorial recursivo no gprof foi de 0.3 e no programa foi de 0.011067. Para 10 milhões de execuções, o gprof mostrou 0.15 segundos ao todo, mas o tempo de relógio da execução foi de 1.082814.

Para o fibonacci recursivo também houve diferença. Por exemplo, para 10 milhões de chamadas, o tempo de relógio foi de 11.509402, mas no gprof deu um total de 1.62 segundos.

Número 20:

```
bin/main fati 20 100000 /tmp/analisefatigprof.out
Answer: 2432902008176640000
Clock time: 0.003696
CPU time: 0.003428 sec user, 0.000187 sec system
gprof bin/main gmon.out > /tmp/analisefatigprof.txt

bin/main fatr 20 100000 /tmp/analisefatrgprof.out
Answer: 2432902008176640000
Clock time: 0.022953
CPU time: 0.021789 sec user, 0.000975 sec system
gprof bin/main gmon.out > /tmp/analisefatrgprof.txt

bin/main fibi 20 100000 /tmp/analisefibigprof.out
Answer: 6765
Clock time: 0.004664
CPU time: 0.004629 sec user, 0.000000 sec system
gprof bin/main gmon.out > /tmp/analisefibigprof.txt

bin/main fibr 20 100000 /tmp/analisefibrgprof.out
Answer: 6765
Clock time: 14.241572
CPU time: 14.202790 sec user, 0.003636 sec system
gprof bin/main gmon.out > /tmp/analisefibrgprof.txt
```

```
bin/main fati 20 10000000 /tmp/analisefatigprof.out
Answer: 2432902008176640000
Clock time: 0.396149
CPU time: 0.361396 sec user, 0.001043 sec system
gprof bin/main gmon.out > /tmp/analisefatigprof.txt

bin/main fatr 20 10000000 /tmp/analisefatrgprof.out
Answer: 2432902008176640000
Clock time: 2.802639
CPU time: 2.288828 sec user, 0.007682 sec system
gprof bin/main gmon.out > /tmp/analisefatrgprof.txt

bin/main fibi 20 10000000 /tmp/analisefibigprof.out
Answer: 6765
Clock time: 0.520196
CPU time: 0.466534 sec user, 0.001982 sec system
gprof bin/main gmon.out > /tmp/analisefibigprof.txt

bin/main fibr 20 10000000 /tmp/analisefibrgprof.out
Answer: 6765
Clock time: 1411.813676
CPU time: 1407.856323 sec user, 0.315970 sec system
gprof bin/main gmon.out > /tmp/analisefibrgprof.txt
```

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  ns/call  ns/call  name
 73.60      0.25     0.25 10000001    25.02    25.02  retornaFatorialRecursivo(int)
 14.72      0.30     0.05                             main
 11.78      0.34     0.04                             retornaFatorialIterativo(int)
  0.00      0.34     0.00        7     0.00     0.00  std::__is_constant_evaluated()
  0.00      0.34     0.00        6     0.00     0.00  std::operator&(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00      0.34     0.00        5     0.00     0.00  std::char_traits<char>::length(char const*)
  0.00      0.34     0.00        3     0.00     0.00  std::__new_allocator<char>::~__new_allocator()
  0.00      0.34     0.00        3     0.00     0.00  void std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*
  0.00      0.34     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::basic_string<std::allocator<cl
  0.00      0.34     0.00        3     0.00     0.00  std::ios_base::setf(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00      0.34     0.00        3     0.00     0.00  std::setprecision(int)
  0.00      0.34     0.00        3     0.00     0.00  std::operator&=(std::_Ios_Fmtflags&, std::_Ios_Fmtflags)
  0.00      0.34     0.00        3     0.00     0.00  std::operator~(std::_Ios_Fmtflags)
  0.00      0.34     0.00        3     0.00     0.00  std::operator|=(std::_Ios_Fmtflags&, std::_Ios_Fmtflags)
  0.00      0.34     0.00        3     0.00     0.00  std::operator|(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00      0.34     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*>(cha
  0.00      0.34     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*>(cha
  0.00      0.34     0.00        2     0.00     0.00  int __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsi
  0.00      0.34     0.00        2     0.00     0.00  std::char_traits<char>::compare(char const*, char const*, unsigned long)
  0.00      0.34     0.00        2     0.00     0.00  std::__cxx11::stoi(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, uns
  0.00      0.34     0.00        2     0.00     0.00  bool std::operator==<char, std::char_traits<char>, std::allocator<char> >(std::__cxx11::basic_string<char, std
  0.00      0.34     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00      0.34     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00      0.34     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00      0.34     0.00        1     0.00     0.00  diffUserTime(rusage*, rusage*)
  0.00      0.34     0.00        1     0.00     0.00  diffSystemTime(rusage*, rusage*)
```

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  us/call  us/call  name
 92.13    180.73   180.73 10000001    18.07    18.07  retornaFibonacciRecursivo(int)
  7.94    196.31    15.57                             retornaFibonacciIterativo(int)
  0.03    196.36     0.05                             main
  0.00    196.36     0.00       11     0.00     0.00  std::__is_constant_evaluated()
  0.00    196.36     0.00        7     0.00     0.00  std::char_traits<char>::length(char const*)
  0.00    196.36     0.00        6     0.00     0.00  std::operator&(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00    196.36     0.00        4     0.00     0.00  std::char_traits<char>::compare(char const*, char const*, unsigned long)
  0.00    196.36     0.00        4     0.00     0.00  bool std::operator==<char, std::char_traits<char>, std::allocator<char> >(std::__cxx11::basic_string<char, std
  0.00    196.36     0.00        3     0.00     0.00  std::__new_allocator<char>::~__new_allocator()
  0.00    196.36     0.00        3     0.00     0.00  void std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*
  0.00    196.36     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::basic_string<std::allocator<cl
  0.00    196.36     0.00        3     0.00     0.00  std::ios_base::setf(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00    196.36     0.00        3     0.00     0.00  std::setprecision(int)
  0.00    196.36     0.00        3     0.00     0.00  std::operator&=(std::_Ios_Fmtflags&, std::_Ios_Fmtflags)
  0.00    196.36     0.00        3     0.00     0.00  std::operator~(std::_Ios_Fmtflags)
  0.00    196.36     0.00        3     0.00     0.00  std::operator|=(std::_Ios_Fmtflags&, std::_Ios_Fmtflags)
  0.00    196.36     0.00        3     0.00     0.00  std::operator|(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00    196.36     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*>(cha
  0.00    196.36     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*>(cha
  0.00    196.36     0.00        2     0.00     0.00  int __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsi
  0.00    196.36     0.00        2     0.00     0.00  std::__cxx11::stoi(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, uns
  0.00    196.36     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00    196.36     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00    196.36     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00    196.36     0.00        1     0.00     0.00  diffUserTime(rusage*, rusage*)
  0.00    196.36     0.00        1     0.00     0.00  diffSystemTime(rusage*, rusage*)
```

Os testes dos algoritmos com o número 20 se tornaram ainda mais demorados na implementação recursiva de fibonacci, o que só mostra o quão

pior o tempo de execução pode aumentar com um incremento de apenas 10 na entrada quando a complexidade é exponencial.

Com 100 mil iterações, a diferença do algoritmo recursivo de fibonacci e do iterativo é de 14 segundos. Logo em seguida, há o teste com 10 milhões de iterações e a diferença se torna aproximadamente 1400 segundos(mais de 23 minutos de tempo de relógio).

O resto das observações, como o tempo de relógio e tempos de usuário e sistemas são semelhantes à bateria de testes anterior com o número 10. A diferença entre o tempo de relógio e o tempo estipulado no gprof também se repete na execução dos testes com o número 20. Os segundos cumulativos totalizam quase 200 segundos, enquanto o tempo de relógio se dá em mais de 1400 segundos, como já citado anteriormente.

Grande parte da análise anterior se repete no caso atual, mas de forma ainda mais visível, como por exemplo a análise do tempo de relógio que deixa claro como as complexidades são diferentes e a importância de como um algoritmo é implementado. Com apenas uma pequena mudança na entrada o resultado foi de 11 segundos com o número 10, para mais de 23 minutos rodando o programa.

Diferentemente das implementações recursivas, a implementação iterativa de ambos programas fatorial e fibonacci se comportaram muito bem, aumentando um pouco sim, mas um valor que não passa de um segundo.

Incremento da função seno nos algoritmos recursivos

A partir de agora, os algoritmos recursivos terão a função seno, que é uma função com um custo computacional significativo, executando 100 vezes nas chamadas(o que pode parecer pouco, mas considerando a quantidade de vezes que os algoritmos recursivos são executados, acaba resultando em uma diferença muito visível do desempenho) para analisar a diferença entre o antes e o depois. Segue os testes abaixo:

```
unsigned long long retornaFatorialRecursivo(int n){
    if(n == 1) return 1;

    for(int i = 0; i < 100; i++) sin(n);
    return n*retornaFatorialRecursivo(n-1);
}
```

```c
int retornaFibonacciRecursivo(int n){
    if(n == 1 || n == 2) return 1;

    for(int i = 0; i < 100; i++) sin(n);
    return retornaFibonacciRecursivo(n-1) + retornaFibonacciRecursivo(n-2);
}
```

Número 10:

```
bin/main fati 10 100000 /tmp/analisefatigprof.out
Answer: 3628800
Clock time: 0.001778
CPU time: 0.001758 sec user, 0.000000 sec system
gprof bin/main gmon.out > /tmp/analisefatigprof.txt

bin/main fatr 10 100000 /tmp/analisefatrgprof.out
Answer: 3628800
Clock time: 0.151693
CPU time: 0.151323 sec user, 0.000000 sec system
gprof bin/main gmon.out > /tmp/analisefatrgprof.txt

bin/main fibi 10 100000 /tmp/analisefibigprof.out
Answer: 55
Clock time: 0.002435
CPU time: 0.002425 sec user, 0.000000 sec system
gprof bin/main gmon.out > /tmp/analisefibigprof.txt

bin/main fibr 10 100000 /tmp/analisefibrgprof.out
Answer: 55
Clock time: 0.911003
CPU time: 0.909180 sec user, 0.000061 sec system
gprof bin/main gmon.out > /tmp/analisefibrgprof.txt
```

```
bin/main fati 10 10000000 /tmp/analisefatigprof.out
Answer: 3628800
Clock time: 0.177779
CPU time: 0.176585 sec user, 0.000031 sec system
gprof bin/main gmon.out > /tmp/analisefatigprof.txt

bin/main fatr 10 10000000 /tmp/analisefatrgprof.out
Answer: 3628800
Clock time: 14.959167
CPU time: 14.910032 sec user, 0.003917 sec system
gprof bin/main gmon.out > /tmp/analisefatrgprof.txt

bin/main fibi 10 10000000 /tmp/analisefibigprof.out
Answer: 55
Clock time: 0.240352
CPU time: 0.239898 sec user, 0.000000 sec system
gprof bin/main gmon.out > /tmp/analisefibigprof.txt

bin/main fibr 10 10000000 /tmp/analisefibrgprof.out
Answer: 55
Clock time: 90.849352
CPU time: 90.563583 sec user, 0.009880 sec system
gprof bin/main gmon.out > /tmp/analisefibrgprof.txt
```

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  us/call  us/call  name
 99.83     13.73    13.73 10000001     1.37     1.37  retornaFatorialRecursivo(int)
  0.15     13.75     0.02                             retornaFatorialIterativo(int)
  0.11     13.76     0.02                             retornaFibonacciIterativo(int)
  0.00     13.76     0.00        7     0.00     0.00  std::__is_constant_evaluated()
  0.00     13.76     0.00        6     0.00     0.00  std::operator&(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00     13.76     0.00        5     0.00     0.00  std::char_traits<char>::length(char const*)
  0.00     13.76     0.00        3     0.00     0.00  std::__new_allocator<char>::~__new_allocator()
  0.00     13.76     0.00        3     0.00     0.00  void std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*
  0.00     13.76     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::basic_string<std::allocator<cl
  0.00     13.76     0.00        3     0.00     0.00  std::ios_base::setf(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00     13.76     0.00        3     0.00     0.00  std::setprecision(int)
  0.00     13.76     0.00        3     0.00     0.00  std::operator&=(std::_Ios_Fmtflags&, std::_Ios_Fmtflags)
  0.00     13.76     0.00        3     0.00     0.00  std::operator~(std::_Ios_Fmtflags)
  0.00     13.76     0.00        3     0.00     0.00  std::operator|=(std::_Ios_Fmtflags&, std::_Ios_Fmtflags)
  0.00     13.76     0.00        3     0.00     0.00  std::operator|(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00     13.76     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*>(cha
  0.00     13.76     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*>(cha
  0.00     13.76     0.00        2     0.00     0.00  int __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsi
  0.00     13.76     0.00        2     0.00     0.00  std::char_traits<char>::compare(char const*, char const*, unsigned long)
  0.00     13.76     0.00        2     0.00     0.00  std::__cxx11::stoi(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, uns
  0.00     13.76     0.00        2     0.00     0.00  bool std::operator==<char, std::char_traits<char>, std::allocator<char> >(std::__cxx11::basic_string<char, std
  0.00     13.76     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00     13.76     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00     13.76     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00     13.76     0.00        1     0.00     0.00  diffUserTime(rusage*, rusage*)
  0.00     13.76     0.00        1     0.00     0.00  diffSystemTime(rusage*, rusage*)
```

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  us/call  us/call  name
 99.84     78.67    78.67 10000001     7.87     7.87  retornaFibonacciRecursivo(int)
  0.22     78.84     0.18                             retornaFibonacciIterativo(int)
  0.03     78.86     0.02                             main
  0.00     78.86     0.00       11     0.00     0.00  std::__is_constant_evaluated()
  0.00     78.86     0.00        7     0.00     0.00  std::char_traits<char>::length(char const*)
  0.00     78.86     0.00        6     0.00     0.00  std::operator&(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00     78.86     0.00        4     0.00     0.00  std::char_traits<char>::compare(char const*, char const*, unsigned long)
  0.00     78.86     0.00        4     0.00     0.00  bool std::operator==<char, std::char_traits<char>, std::allocator<char> >(std::__cxx11::basic_string<char, std
  0.00     78.86     0.00        3     0.00     0.00  std::__new_allocator<char>::~__new_allocator()
  0.00     78.86     0.00        3     0.00     0.00  void std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*
  0.00     78.86     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::basic_string<std::allocator<cl
  0.00     78.86     0.00        3     0.00     0.00  std::ios_base::setf(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00     78.86     0.00        3     0.00     0.00  std::setprecision(int)
  0.00     78.86     0.00        3     0.00     0.00  std::operator&=(std::_Ios_Fmtflags&, std::_Ios_Fmtflags)
  0.00     78.86     0.00        3     0.00     0.00  std::operator~(std::_Ios_Fmtflags)
  0.00     78.86     0.00        3     0.00     0.00  std::operator|=(std::_Ios_Fmtflags&, std::_Ios_Fmtflags)
  0.00     78.86     0.00        3     0.00     0.00  std::operator|(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00     78.86     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*>(cha
  0.00     78.86     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*>(cha
  0.00     78.86     0.00        2     0.00     0.00  int __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsi
  0.00     78.86     0.00        2     0.00     0.00  std::__cxx11::stoi(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, uns
  0.00     78.86     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00     78.86     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00     78.86     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00     78.86     0.00        1     0.00     0.00  diffUserTime(rusage*, rusage*)
  0.00     78.86     0.00        1     0.00     0.00  diffSystemTime(rusage*, rusage*)
```

Número 20:

```
bin/main fati 20 10000000 /tmp/analisefatigprof.out
Answer: 2432902008176640000
Clock time: 0.353570
CPU time: 0.353154 sec user, 0.000000 sec system
gprof bin/main gmon.out > /tmp/analisefatigprof.txt

bin/main fatr 20 10000000 /tmp/analisefatrgprof.out
Answer: 2432902008176640000
Clock time: 31.019542
CPU time: 30.986967 sec user, 0.000139 sec system
gprof bin/main gmon.out > /tmp/analisefatrgprof.txt

bin/main fibi 20 10000000 /tmp/analisefibigprof.out
Answer: 6765
Clock time: 0.449983
CPU time: 0.449336 sec user, 0.000088 sec system
gprof bin/main gmon.out > /tmp/analisefibigprof.txt

bin/main fibr 20 10000000 /tmp/analisefibrgprof.out
Answer: 6765
Clock time: 11286.293392
CPU time: 11257.792969 sec user, 2.536113 sec system
gprof bin/main gmon.out > /tmp/analisefibrgprof.txt
```

```
bin/main fati 20 100000 /tmp/analisefatigprof.out
Answer: 2432902008176640000
Clock time: 0.004314
CPU time: 0.003767 sec user, 0.000000 sec system
gprof bin/main gmon.out > /tmp/analisefatigprof.txt

bin/main fatr 20 100000 /tmp/analisefatrgprof.out
Answer: 2432902008176640000
Clock time: 0.317741
CPU time: 0.317051 sec user, 0.000000 sec system
gprof bin/main gmon.out > /tmp/analisefatrgprof.txt

bin/main fibi 20 100000 /tmp/analisefibigprof.out
Answer: 6765
Clock time: 0.004853
CPU time: 0.004771 sec user, 0.000000 sec system
gprof bin/main gmon.out > /tmp/analisefibigprof.txt

bin/main fibr 20 100000 /tmp/analisefibrgprof.out
Answer: 6765
Clock time: 114.654703
CPU time: 113.684792 sec user, 0.062968 sec system
gprof bin/main gmon.out > /tmp/analisefibrgprof.txt
```

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  us/call  us/call  name
 99.57     28.58    28.58 10000001     2.86     2.86  retornaFatorialRecursivo(int)
  0.35     28.68     0.10                             retornaFatorialIterativo(int)
  0.10     28.71     0.03                             retornaFibonacciIterativo(int)
  0.07     28.73     0.02                             main
  0.00     28.73     0.00        7     0.00     0.00  std::__is_constant_evaluated()
  0.00     28.73     0.00        6     0.00     0.00  std::operator&(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00     28.73     0.00        5     0.00     0.00  std::char_traits<char>::length(char const*)
  0.00     28.73     0.00        3     0.00     0.00  std::__new_allocator<char>::~__new_allocator()
  0.00     28.73     0.00        3     0.00     0.00  void std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*
  0.00     28.73     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::basic_string<std::allocator<cl
  0.00     28.73     0.00        3     0.00     0.00  std::ios_base::setf(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00     28.73     0.00        3     0.00     0.00  std::setprecision(int)
  0.00     28.73     0.00        3     0.00     0.00  std::operator&=(std::_Ios_Fmtflags&, std::_Ios_Fmtflags)
  0.00     28.73     0.00        3     0.00     0.00  std::operator~(std::_Ios_Fmtflags)
  0.00     28.73     0.00        3     0.00     0.00  std::operator|=(std::_Ios_Fmtflags&, std::_Ios_Fmtflags)
  0.00     28.73     0.00        3     0.00     0.00  std::operator|(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00     28.73     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*>(cha
  0.00     28.73     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*>(cha
  0.00     28.73     0.00        2     0.00     0.00  int __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsi
  0.00     28.73     0.00        2     0.00     0.00  std::char_traits<char>::compare(char const*, char const*, unsigned long)
  0.00     28.73     0.00        2     0.00     0.00  std::__cxx11::stoi(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, uns
  0.00     28.73     0.00        2     0.00     0.00  bool std::operator==<char, std::char_traits<char>, std::allocator<char> >(std::__cxx11::basic_string<char, std
  0.00     28.73     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00     28.73     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00     28.73     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00     28.73     0.00        1     0.00     0.00  diffUserTime(rusage*, rusage*)
  0.00     28.73     0.00        1     0.00     0.00  diffSystemTime(rusage*, rusage*)
```

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  us/call  us/call  name
 98.12   1301.37  1301.37 10000001   130.14   130.14  retornaFibonacciRecursivo(int)
  1.95   1327.26    25.88                             retornaFibonacciIterativo(int)
  0.01   1327.44     0.18                             main
  0.00   1327.44     0.00       11     0.00     0.00  std::__is_constant_evaluated()
  0.00   1327.44     0.00        7     0.00     0.00  std::char_traits<char>::length(char const*)
  0.00   1327.44     0.00        6     0.00     0.00  std::operator&(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00   1327.44     0.00        4     0.00     0.00  std::char_traits<char>::compare(char const*, char const*, unsigned long)
  0.00   1327.44     0.00        4     0.00     0.00  bool std::operator==<char, std::char_traits<char>, std::allocator<char> >(std::__cxx11::basic_string<char, std
  0.00   1327.44     0.00        3     0.00     0.00  std::__new_allocator<char>::~__new_allocator()
  0.00   1327.44     0.00        3     0.00     0.00  void std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*
  0.00   1327.44     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::basic_string<std::allocator<c
  0.00   1327.44     0.00        3     0.00     0.00  std::ios_base::setf(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00   1327.44     0.00        3     0.00     0.00  std::setprecision(int)
  0.00   1327.44     0.00        3     0.00     0.00  std::operator&=(std::_Ios_Fmtflags&, std::_Ios_Fmtflags)
  0.00   1327.44     0.00        3     0.00     0.00  std::operator~(std::_Ios_Fmtflags)
  0.00   1327.44     0.00        3     0.00     0.00  std::operator|=(std::_Ios_Fmtflags&, std::_Ios_Fmtflags)
  0.00   1327.44     0.00        3     0.00     0.00  std::operator|(std::_Ios_Fmtflags, std::_Ios_Fmtflags)
  0.00   1327.44     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*>(cha
  0.00   1327.44     0.00        3     0.00     0.00  std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char const*>(cha
  0.00   1327.44     0.00        2     0.00     0.00  int __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsi
  0.00   1327.44     0.00        2     0.00     0.00  std::__cxx11::stoi(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, uns
  0.00   1327.44     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00   1327.44     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00   1327.44     0.00        2     0.00     0.00  __gnu_cxx::__stoa<long, int, char, int>(long (*)(char const*, char**, int), char const*, char const*, unsigned
  0.00   1327.44     0.00        1     0.00     0.00  diffUserTime(rusage*, rusage*)
  0.00   1327.44     0.00        1     0.00     0.00  diffSystemTime(rusage*, rusage*)
```

Com o incremento de 100 iterações da função seno nos algoritmos recursivos a discrepância do tempo de execução do programa para antes ficou imensa.

Tomando o 20 como exemplo, com 100 mil iterações o tempo passou de 14 segundos para quase 2 minutos. Apesar disso, o que mais chama atenção é a execução com 10 milhões de iterações. De um pouco mais de 20 minutos, o programa demorou mais de 3 horas para dar um resultado. Daí podemos ver como um algoritmo de complexidade exponencial é ruim se falado sobre o tempo gasto para que seja completado.

Com relação às outras medições, as observações são parecidas, tempo de relógio e tempo acumulado do gprof ficaram diferentes(gprof sempre estando menor do que o tempo de relógio). Tempo de usuário acompanhando o tempo de relógio e o tempo de sistema não muito distante do 0.

Conclusão

Com a bateria de testes realizados e dados analisados, foi possível notar a diferença de tempo entre algoritmos recursivos e iterativos e correlacionar essas informações à complexidade computacional dos algoritmos já citados anteriormente aos testes.

É possível notar os algoritmos iterativos se destacando em performance mesmo com um aumento de iterações dos algoritmos. Mesmo rodando 10 milhões de vezes as mesmas instruções, não houveram grandes mudanças de desempenho.

Em contrapartida, os algoritmos recursivos tiveram mudanças mais perceptíveis, principalmente com o incremento da função seno. O fatorial com complexidade O(n) teve uma piora de, aproximadamente, 28 segundos ao incrementar o cálculo do seno. Já o algoritmo para fibonacci teve uma piora de mais de 2 horas, fazendo jus à sua complexidade exponencial.

Dada toda a análise, torna-se mais clara a importância de analisar as possíveis soluções para o mesmo problema e pensar em formas eficientes de resolução. Mesmo que dois algoritmos deem a mesma resposta, eles se comportam de maneiras diferentes e dependendo do tamanho da entrada de um problema, a solução pode se mostrar ruim, tanto pelo tempo gasto para execução, como também seu custo computacional.