



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Dissertação em Trabalho de Conclusão de Curso (TCC) –
Bacharelado em Ciências da Computação.

Roberto Carlos dos Santos

***RecPy*: pré-compilador para estudo da conversão de
funções recursivas**

Orientador: Professor Dr. Fabiano de Souza Oliveira

Rio de Janeiro
2021



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

RESUMO

- conceitos básicos relacionados a funções recursivas;
- Benefícios das conversões, entre si, de *algoritmos recursivos ou iterativos*;
- o aplicativo *RecPy*;
- resultados obtidos;
- conclusões gerais do trabalho.



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Conceitos úteis – Função recursiva

Contém uma ou várias chamadas a si mesma.

Pode ser de dois tipos:

- recursiva caudal (TR) ou
- recursiva não caudal (NTR).



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Conceitos úteis – Função recursiva não caudal (Non Tail Recursion-NTR)

Função que, para retornar resultados,
depende de operação extra.

Exemplo de função recursiva não caudal:

```
def fatNTR(n):  
    if n < 2:  
        return 1  
    return fatNTR(n - 1) * n
```



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Recursão não caudal (Non Tail Recursion-NTR)

Exemplo de estrutura de pilha

← → ↻ ⚠ Não seguro | www.pythontutor.com/visualize.html#mode=display

[Get live help](#) for free in the [Python tutoring Discord](#) chat room

Python 2.7
([known limitations](#))

```
→ 1 def fatNTR(n):  
  2     if n < 2:  
  3         return 1  
→ 4     return n * fatNTR(n - 1)  
  5  
  6 print(fatNTR(3))
```

[Edit this code](#)

→ line that just executed
→ next line to execute

Step 9 of 14

[Customize visualization](#) (NEW!)

Print output (drag lower right corner to resize)

Frames	Objects
Global frame fatNTR	function fatNTR(n)
fatNTR n	int 3
fatNTR n	int 2
fatNTR n	int 1



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Conceitos úteis – Recursão caudal (Tail Recursion-TR)

- A função retorna seu resultado sem necessidade de outra operação.
- Mais eficiente que as NTR.
- Dispensa um quadro de pilha novo para cada chamada recursiva.

Exemplo de função recursiva caudal resultante
da conversão NTR-TR, no RecPy:

```
def fatTR(n, acc=1):  
    if n < 2:  
        return acc  
    return fatTR(n - 1, n * acc)
```



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Recursão caudal (Tail Recursion-TR)

Exemplo de estrutura de pilha não otimizada

← → ↻ ⚠ Não seguro | www.pythontutor.com/visualize.html#mode=display

[Get live help](#) for free in the [Python tutoring Discord](#) chat room

Python 2.7
([known limitations](#))

```
→ 1 def fatTR(n, acc=1):  
  2     if n < 2:  
  3         return acc  
→ 4     return fatTR(n - 1, n * acc)  
  5 print(fatTR(3))
```

[Edit this code](#)

→ line that just executed
→ next line to execute

Step 9 of 14

[Customize visualization](#) (NEW!)

Print output (drag lower right corner to resize)

Frames Objects

Global frame
fatTR

fatTR
n
acc

fatTR
n
acc

fatTR
n
acc

function fatTR(n, acc)
default arguments:
acc

int 1

int 3

int 2

int 6



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Conceitos úteis – Eliminação da chamada de cauda (Tail Recursion Elimination-TRE ou Tail Call Elimination-TCE)

O RecPy elimina a chamada de cauda transformando-a em uma função iterativa.

Exemplo de função iterativa resultante da conversão NTR-IT,
no RecPy:

```
def fatIT(n, acc=1):  
    while not n < 2:  
        n, acc = n - 1, n * acc  
    return acc
```




Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Função iterativa (IT)

Exemplo de estrutura de pilha

← → ↻ ⚠ Não seguro | www.pythontutor.com/visualize.html#mode=display

[Get live help](#) for free in the [Python tutoring Discord](#) chat room

```
Python 2.7  
(known limitations)  
1 def fatIT(n, acc=1):  
→ 2     while not n < 2:  
→ 3         n, acc = n - 1, n * acc  
4     return acc  
5  
6 print(fatIT(3))
```

[Edit this code](#)

→ line that just executed
→ next line to execute

Step 8 of 10

[Customize visualization](#) (NEW!)

Print output (drag lower right corner to resize)

Frames

Global frame

fatIT

Objects

function fatIT(n, acc)
default arguments:
acc

int 1

int 6

n

acc



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Conceitos úteis – Indução matemática

Prova a validade de fórmulas conjecturadas ou deduzidas.

Forte correlação entre os algoritmos recursivos e a
noção de indução matemática.



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Conceitos úteis – Algoritmos de divisão e conquista

Resolve um problema:

1. Quebrando-o em instâncias menores de mesmo tipo;
2. Recursivamente resolvendo esses subproblemas; e
3. Combinando suas respostas apropriadamente.



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Objetivo geral

Oferecer à comunidade científica e acadêmica aplicação destinada ao estudo sistemático de técnicas de conversão de algoritmos.



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Objetivos específicos

Abordar:

- Vantagens e desvantagens de algoritmos recursivos;
- Comparação, em termos de eficiência, de algoritmos;
- Problemas que podem ocorrer nos algoritmos recursivos;
 - Apresentação de exemplos no *RecPy*;
- restrições sintáticas na linguagem reconhecida pelo *RecPy*.
 - Descrição do desenvolvimento do *RecPy*.



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Motivação e justificativa do trabalho

Colaborar com
o ensino e o aprendizado de conversão de funções
recursivas.



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Levantamento de hipóteses

- Algumas linguagens, como Python, apresentam problemas com códigos recursivos.
- Há benefícios na transformação de uma função recursiva em iterativa.
- Há vantagens das rotinas recursivas caudais em relação às não caudais, na otimização dos códigos.
- Há viabilidade e benefícios, em termos de ganhos de eficiência ou de facilitação de estudos, na tradução automática de rotinas recursivas.



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

REVISÃO DA LITERATURA CIENTÍFICA

Diferenças em relação a outros trabalhos que tratam da conversão de funções recursivas

- Abordagem eminentemente prática, sem descuidar de aspectos teóricos;
- Não se restringe à temática de otimização de funções recursivas. Trata, também, de questões de interesse no estudo da disciplina de Compiladores.



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

CONVERSÃO AUTOMÁTICA DE FUNÇÕES RECURSIVAS: APLICATIVO RECPY

Python é uma dessas linguagens que, apesar de ter inúmeros bons atributos, tem, em sua concepção, a discordância em relação à eliminação das recursões caudais. Por isso mesmo, foi a escolhida como linguagem objeto *RecPy*, destinado ao estudo e automatização de conversões entre algoritmos recursivos e iterativos.



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Vantagens de algoritmos recursivos

- Poderoso método de resolução de problemas científicos;
- Frequentemente produzem soluções elegantes e mais fáceis de entender do que as respectivas iterativas;
- Frequentemente torna mais claros programas complexos;
- Solução natural em implementações que envolvam manipulações de árvores;
- Permite resolução de problemas através do princípio de dividir e conquistar;
- São comumente mais concisos;
- Possibilidade do uso de induções matemáticas para comprovação de seu funcionamento correto.



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Desvantagens da utilização de algoritmos recursivos

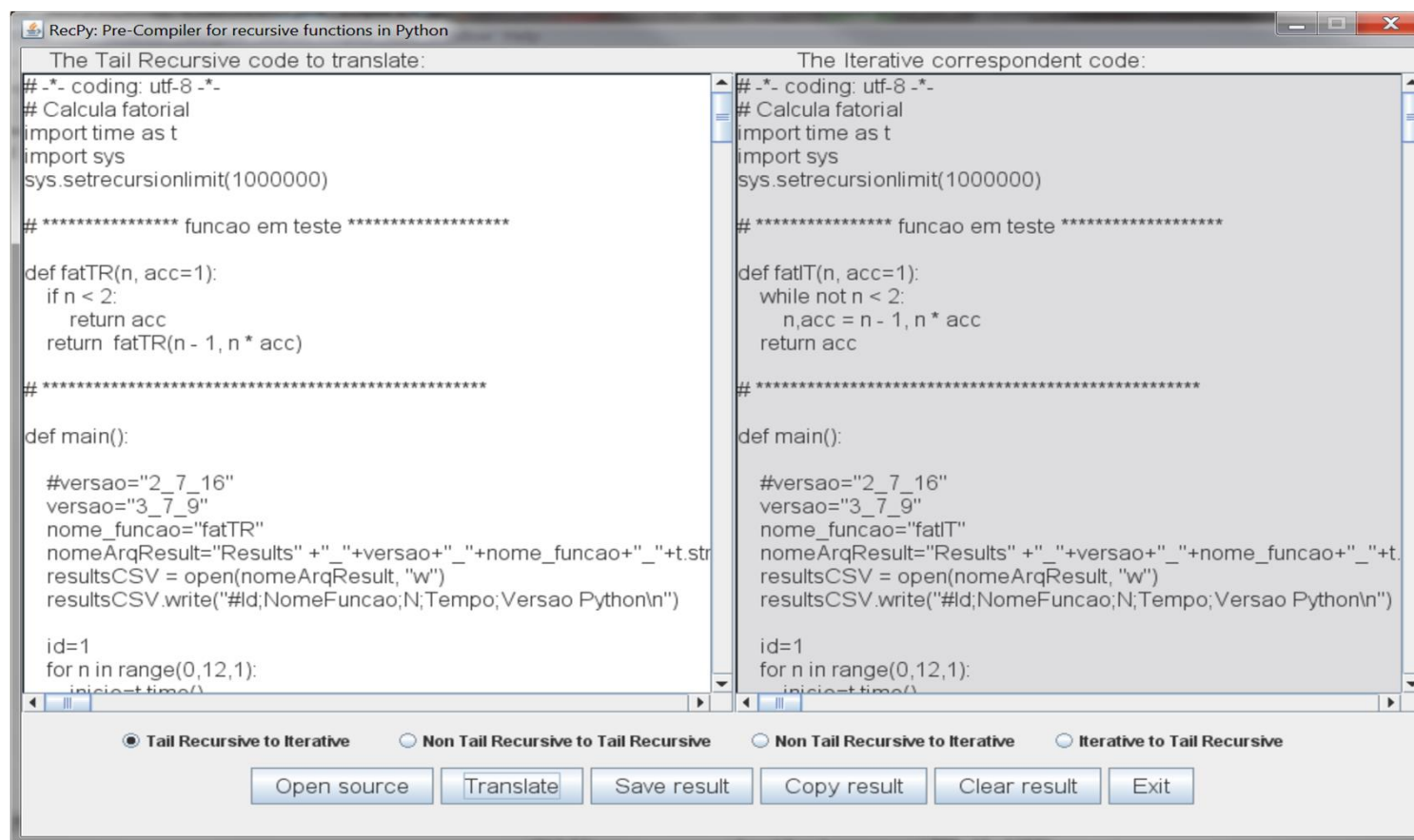
- Aumentam o uso intensivo da memória de pilha;
- Tendem a ser mais lentos do que os equivalentes iterativos;
- Especialmente quando há repetição de subproblemas, tornam-se ineficientes;
- Propensos ao estouro de pilha;
- São mais difíceis de serem depurados.



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Apresentação do *RecPy*





Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Apresentação do *RecPy*

O RecPy:

- simplifica a conversão de algoritmos recursivos;
- Facilita o estudo de funções recursivas;
- serve potencialmente como material de estudo para a disciplina de Compiladores.



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Escopo de conversões possíveis no RecPy

O *RecPy* realiza quatro tipos de conversão específicos, com as especificações e restrições apresentadas no Item 5.5 do TCC:

- 1) a conversão de função recursiva caudal em função iterativa (TR-IT);
- 2) a conversão de função recursiva não caudal em função recursiva caudal (NTR-TR);
- 3) a conversão de função recursiva não caudal em função iterativa (NTR-IT);
- 4) a conversão de função iterativa em função recursiva caudal (IT-TR).



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Como utilizar o RecPy

Link para o vídeo de orientações para a utilização do RecPy:

https://youtu.be/D_fMF0I5OeM



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Formatos de funções recursivas reconhecíveis no *RecPy*

1) Funções recursivas caudais simples:

```
def <nome_da_funcao_recursiva_caudal>( <lista de parâmetros 1>):  
    <bloco de códigos intermediários ou comentários>  
    if <condição>:  
        <bloco de códigos intermediários ou comentários>  
        return <variável_de_retorno>  
    else:  
        <bloco de códigos intermediários ou comentários>  
        return <nome_da_funcao_recursiva_caudal>( <lista de parâmetros 2>)
```




Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Formatos de funções recursivas reconhecíveis no *RecPy*

2) Funções recursivas não caudais simples:

```
def <nome_da_funcao_recursiva_nao_caudal>( <lista de parâmetros 1>):  
    <bloco de códigos intermediários ou comentários>  
    if <condição>:  
        <bloco de códigos intermediários ou comentários>  
        return <variável_de_retorno>  
    else:  
        <bloco de códigos intermediários ou comentários>  
        return <expr> <operador> <nome_da_funcao_recursiva_nao_caudal>( <lista de  
parâmetros 2>) OU return <nome_da_funcao_recursiva_nao_caudal>( <lista de parâmetros 2>)  
    <operador> <expr>
```



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Formatos de funções recursivas reconhecíveis no *RecPy*

3) Funções iterativas simples:

```
def <nome_da_funcao_iterativa>(<lista de parâmetros 1>):  
    <bloco de códigos intermediários ou comentários>  
    while <condição>:  
        <bloco de códigos intermediários ou comentários>  
        <lista de parametros1> = <lista de parametros 2>  
    return <expressao_de_retorno>
```



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Resumo dos principais Requisitos, restrições, observações e recomendações

1. Utilize padrões reconhecíveis pelo *RecPy*: Tópico 5.4.;
2. Na nomenclatura das funções, utilize os prefixos (ou sufixos) IT, TR ou NTR;
3. Na tradução TR-IT, mantenha a mesma ordem das variáveis contidas em listas;
4. Observe os cuidados com os IFs, WHILEs e RETURNS;
5. Evite parêntesis desnecessários;
6. Utilize a sintaxe de Python 2.7.xx;
7. Mantenha a primeira linha (def ...) junto à margem esquerda;
8. Cuidado com comentários que contenham palavras reservadas do Python;
9. Em caso de erros, experimente retirar os comentários;
10. Quando experimentar uma função isolada, insira uma linha em branco ao final;
11. Na tradução IT-TR, apenas uma linha do bloco da função deve conter o caractere “=”.



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Desenvolvimento do aplicativo *RecPy*

O ANTLR4 foi escolhido para desenvolvimento do núcleo do *RecPy*, em razão de suas características:

- possibilitar desenvolvimento rápido;
- facilidade de aprendizado;
- simplicidade de alterações de códigos do compilador.
- a gramática pode ser ampliada para outros casos e linguagens de modo rápido e simples.
- Na versão atual – ANTLR 4.9.1, a ferramenta pode gerar código em Java, C#, Python 2 e 3, Javascript, Go, C++, Swift e PHP.



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Desenvolvimento do núcleo do RecPy, no Antlr4

Link para o vídeo de orientações sobre o Desenvolvimento do núcleo de conversão do RecPy, no Antlr4.

Inclui geração de tokens e de árvores sintáticas abstratas (AST), bem como a codificação do *parser* e do *lexer utilizados no núcleo de conversão do RecPy*:

<https://youtu.be/NhMietucQFI>



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Experimentos realizados

Resultados gráficos contidos na pasta Apresentação de
resultados > Gráficos de resultados



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Síntese dos resultados

Quanto ao sucesso no procedimento de conversão e reconhecimento do algoritmo

Algoritmo	Versão original	TR->IT: a conversão funcionou?	NTR->TR: a conversão funcionou?	NTR->IT: a conversão funcionou?	IT->TR: a conversão funcionou?
<u>FatMod</u>	<u>fatModNTR</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Fatorial</u>	<u>fatNTR</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Fibonacci</u>	<u>fibTR</u>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Inverte String</u>	<u>invNTR</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>MDC</u>	<u>mdcTR</u>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Ordenação</u>	<u>ordNTR</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Palíndromo</u>	<u>palTR</u>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Potenciação</u>	<u>potNTR</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Primo</u>	<u>primTR</u>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Soma de vetor</u>	<u>sumVecNTR</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<u>Texto</u>	<u>textNTR</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Quadro 8 - Síntese de resultados quanto ao sucesso no procedimento de conversão e de reconhecimento das funções testadas

Legendas:

☒ Sim, funcionou perfeitamente e de modo automático

☐ Não houve versão NTR



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Síntese dos resultados

Quanto ao valor máximo da variável N (quantidade de chamadas recursivas ou de iterações)

Tabela 1 - Resultados quanto ao valor máximo da variável N (quantidade de chamadas recursivas ou de iterações)

Algoritmo	Iterativo (IT) Máx. valor N Python 2.7.16	Recursivo caudal (TR) Máx. valor N Python 2.7.16	Recursivo não caudal (NTR) Máx. valor N Python 2.7.16	Iterativo (IT) Máx. valor N Python 3.7.9	Recursão Caudal (TR) Máx. valor N Python 3.7.9	Recursão não Caudal (NTR) Máx. valor N Python 3.7.9
<u>FatMod</u>	> 9990	3060	4180	> 9990	2160	2610
Fatorial	> 108 mil	3064	4188	>120 mil	2163	2616
Fibonacci	> 9980	4185	□	> 9980	2615	□
<u>Inverte String</u>	> 31 mil	3063	4186	> 30 mil	2162	2613
MDC	> 9986	> 9986	□	> 9986	>9986	□
Ordenação	> 28	> 28	> 30	> 28	> 28	> 28
Palíndromo	> 100 mil	> 100 mil	□	> 100 mil	> 100 mil	□
Potenciação	> 310 mil	2615	4185	> 270 mil	2611	2611
Primo	> 99 mil	> 99 mil	□	> 99 mil	> 99 mil	□
Soma de vetor	> 9990	3060	4180	> 9990	2160	2610
Texto	>3710	3060	4180	>3070	2160	2610

Legendas:

> ##### Maior do que N chamadas recursivas ou iterações (onde ##### é o valor de N). Ou seja, nos casos em que há o sinal ">" na frente, o algoritmo foi experimentado até N chamadas recursivas ou iterações, mas poderia ir além.

□ Não houve versão NTR



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Síntese dos resultados

Quanto à complexidade de tempo de execução dos algoritmos das funções recursivas experimentadas, na notação assintótica

Complexidade de tempo de execução dos algoritmos experimentados

Algoritmo	Versão original	Iterativo (IT)	Recursivo caudal (TR)	Recursivo não caudal (NTR)
<u>FatMod</u>	<u>fatModNTR</u>	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
Fatorial	<u>fatNTR</u>	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
Fibonacci	<u>fibTR</u>	$\Theta(n)$	$\Theta(n)$	\square
<u>Inverte String</u>	<u>invNTR</u>	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
MDC	<u>mdcTR</u>	$O(\log n)$	$O(\log n)$	\square
Ordenação	<u>ordNTR</u>	$O(n^2)$	$O(n^2)$	$O(n^2)$
Palíndromo	<u>palTR</u>	$\Theta(n)$	$\Theta(n)$	\square
Potenciação	<u>potNTR</u>	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
Primo	<u>primTR</u>	$\Theta(n)$	$\Theta(n)$	\square
Soma de vetor	<u>sumVecNTR</u>	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
Texto	<u>textNTR</u>	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$

Quadro 9 - Complexidade de tempo de execução dos algoritmos experimentados



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Resumo das conclusões quanto às hipóteses levantadas

- a) De fato, em alguns casos, os algoritmos recursivos provocam o estouro de pilha;
- b) Na conversão de funções recursivas para iterativas, ficou bem claro o aumento na escalabilidade da quantidade de iterações, nos casos em que as funções recursivas estavam apresentando limitações em suas versões recursivas. Em nenhum dos experimentos as versões iterativas apresentaram limitações;
- c) Em termos de tempo de execução dos códigos, não se vislumbraram diferenças muito significativas entre as versões TR e NTR.
- d) O RecPy mostrou-se ferramenta útil tanto para agilizar o procedimento de conversão de funções recursivas e iterativas quanto no estudo dessas funções.



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

SUGESTÃO DE TRABALHOS FUTUROS

- podem-se utilizar linguagens-alvo diferentes de Python ou linguagens de implementação diferentes de Java.
- Pode-se, também, ampliar o rol de funções recursivas e iterativas reconhecíveis.
- Para fins exclusivamente teóricos e didáticos, podem ser elaborados módulos de conversão IT-NTR ou TR-NTR, embora de pouca utilidade prática.



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências
Instituto de Matemática e Estatística

Agradeço pela sua atenção.

Roberto Carlos dos Santos
roberto.c.santos.rj@gmail.com