Tarefa 1 — Introdução & Fundamentos Matemáticos & Computação e Representação

Teoria da Computação — PPComp

Prof. Jefferson O. Andrade

Ifes — Campus Serra

2021/2

1 Introdução

Para esta tarefa você deve resolver os problemas da Seção 3. A sua resposta deve ser preparada em um documento em LaTeX. A classe do documento deve ser `scrartcl`, com fonte tamanho 11. Cada exercício deve ser respondido em uma seção própria. Caso o exercício possua itens/partes, cada item/parte deve ser respondido em uma subseção.

Para aqueles que não tiverem experiência com LaTeX, sugere-se o uso do ambiente web Overleaf, porque não requer instalação. Sugere-se também os vídeos da *playlist* Introdução ao LaTeX (Curso Completo). Apenas o arquivo PDF gerado pelo LaTeX deve ser entregue como resposta da atividade.

Para a geração de gráficos, pode ser usado um software de escritório como o LibreOffice Calc ou o Microsoft Office Excel, ou algum outro software mais técnico como GnuPlot (*playlist*), ou mesmo gerar os gráficos usando Python e matplotlib (vídeo), por exemplo. Salve os gráficos em um arquivo de formato de imagem e depois inclua os gráficos no seu documento LaTeX.

A qualidade da formatação do seu relatório será um dos fatores considerados na correção. O LaTeX permite a criação de documentos extremamente elegantes, use o potencial da ferramenta.

2 Orientações

Colaboração: você pode colaborar com outros alunos que estão atualmente matriculados nesta disciplina em *brainstorming* e pensando em abordagens para soluções, mas você deve escrever as soluções por conta própria e não pode compartilhá-las com outros alunos.

Violações graves: compartilhar perguntas ou soluções com qualquer pessoa fora desta disciplina, incluindo postagem em sites externos, constitui uma violação do código de ética. Em particular, você não pode obter ajuda de alunos ou materiais de anos anteriores desta disciplina ou equivalente. Casos de **plágio** serão relatados à coordenação de curso e encaminhados ao conselho de ética.

Formato de submissão: O PDF submetido deve ser digitado no mesmo formato que este. Inclua o texto dos problemas e escreva "Solução X" antes da sua solução. Poderão ser deduzidos pontos se você enviar em um formato diferente.

3 Problemas

Problema 1 (*35pts*)

Parte I

Implemente os algoritmos de multiplicação tradicional (aquele que aprendemos na escola) e também o método de multiplicação de de Karatsuba (Algoritmo 0.4 do ITCS) como funções na linguagem Clojure.[1] Suas funções receberão dois vetores com os dígitos dos número que devem ser multiplicados no esquema de dígito menos significante primeiro, i.e, o dígito de ordem 0 estará na posição de índice 0 do vetor. Suas funções devem, igualmente, retornar o resultado da multiplicação como um vetor contento os dígitos do resultado no esquema de dígito menos significante primeiro.

Por exemplo, $534826477703723139155890 9 \times 304788237732936076348730 8$ seria calculado pelas chamadas de função abaixo.

```
(karatsuba-multiply
  [9 0 9 8 5 5 1 9 3 1 3 2 7 3 0 7 7 7 4 6 2 8 4 3 5]
  [8 0 3 7 8 4 3 6 7 0 6 3 9 2 3 7 7 3 2 8 8 7 4 0 3])

(standard-multiply
  [9 0 9 8 5 5 1 9 3 1 3 2 7 3 0 7 7 7 4 6 2 8 4 3 5]
  [8 0 3 7 8 4 3 6 7 0 6 3 9 2 3 7 7 3 2 8 8 7 4 0 3])
```

O resultado da execução das funções seria:

```
[2 7 9 6 2 8 5 5 3 5 4 6 2 8 1 6 4 5 6 4 9 0 2 9 4 4 8 0 0 5 1 0 4 0 2 1 3 2 2 3 6 9
    1 8 8 0 0 3 6 1]
```

Que representa o número 16300881963223120401500844920946546182645355826972.

Parte II

Meça o tempo médio de execução das duas funções para números com quantidades crescentes de dígitos. Plote o gráfico com os tempos de execução das duas funções e determine a partir de que ponto – supondo que seja o caso – a multiplicação de Karatsuba se torna mais rápida que a multiplicação padrão.

Problema 2 (*10pts*)

Use os quantificadores lógicos $\forall$ e $\exists$, bem como os operadores lógicos $\wedge$, $\vee$ e $\neg$, os operadores aritméticos $+$ e $\times$, e os operadores relacionais $=$, $>$ e $<$ para escrever o seguinte:

1. Uma expressão $\phi(n, m)$ tal que, para quaisquer números naturais n e m, $\phi(n, m)$ é verdadeiro se e somente se n e m são primos entre si.
2. Uma expressão $\phi(n)$ tal que para todo número natural n, $\phi(n)$ seja verdadeiro se e somente se n for uma potência de cinco.

Problema 3 (*15pts*)

Seja $n \in \mathbb{N}$. Para cada um dos seguintes pares de conjuntos (S, T), prove ou refute a seguinte afirmação: existe uma função f, *injetora*[2], mapeando S em T.

[1] Para incluir o código Clojure em sua resposta, utilize o pacote `listings` do LaTeX, com as configurações específicas para Clojure disponíveis aqui: `https://alexott.blogspot.com/2010/01/clojure-latex.html`.

[2] Em português, o termo *um-para-um* tipicamente se refere a funções **bijetoras**, mas em inglês o termo equivalente *one-to-one* normalmente se refere a funções **injetoras**, o que causa alguma confusão, e faz com que pessoas que estão mais acostumada à terminologia em inglês acabem utilizando o termo *um-para-um* como se significasse *injetora* mesmo em português. Por este motivo, vou evitar o uso dos termos *um-para-um* e *one-to-one*, e dar preferência aos termos termos mais formais (e menos ambíguos) *injetora* e *bijetora*, conforme o caso.

1. Seja $n > 10$. $S = \{0,1\}^n$ e $T = [n] \times [n] \times [n]$.[3]
2. Seja $n > 10$. S é o conjunto de todas as funções mapeando $\{0,1\}^n$ para $\{0,1\}$. $T = \{0,1\}^{n^3}$.
3. Seja $n > 100$. $S = \{k \in [n] : k \text{ é primo}\}$, $T = \{0,1\}^{\lceil \log n - 1 \rceil}$.

Problema 4 (*20pts*)

A codificação ASCII pode ser usada para codificar cadeias de n letras em inglês como cadeias binárias de $7n$ bits. Nesse exercício é pedido que você encontre uma codificação mais compacta para cadeias de letras minúsculas em inglês.

1. Prove que existe um esquema de representação (E, D) para strings sobre o alfabeto de 26 letras $\{a, b, c, \ldots, x, y, z\}$ como cadeias binárias tal que para todo $n > 0$ e cadeia $x \in \{a, b, c, \ldots, x, y, z\}^n$, de tamanho n, a representação $E(x)$ é uma cadeia binária de tamanho máximo $4.8n + 1000$. Em outras palavras, prove que para todo n, existe uma função injetora $E : \{a, b, c, \ldots, x, y, z\}^n \to \{0,1\}^{\lceil 4.8n+1000 \rceil}$.
2. Prove que **não existe** esquema de representação sobre o alfabeto $\{a, b, c, \ldots, x, y, z\}$ como cadeias binárias tal que para toda cadeia $x \in \{a, b, c, \ldots, x, y, z\}^n$, de tamanho n, a representação $E(x)$ seja uma cadeia binária de tamanho $\lceil 4.6n + 1000 \rceil$. Em outras palavras, prove que existe algum $n > 0$ tal que não existe função injetora $E : \{a, b, c, \ldots, x, y, z\}^n \to \{0,1\}^{\lceil 4.6n+1000 \rceil}$.

Problema 5 (*20pts*)

1. Foi demonstrado que os números naturais podem ser representados como cadeias binárias. Prove que o inverso também se aplica, i.e., que existe uma função injetora $StN : \{0,1\}^\star \to \mathbb{N}$.[4]
2. O Teorema de Canto provou que não existe uma função injetora $RtN : \mathbb{R} \to \mathbb{N}$. Demonstre que o Teorema de Canto implica no teorema abaixo:

   Teorema: Não existe uma função injetora $RtS : \mathbb{R} \to \{0,1\}^\star$.

[3] Lembre-se que definiu-se que a notação $[n]$ representa um conjunto de números naturais tal que $[n] = \{k : k \in \mathbb{N} \wedge k < n\}$, ou seja, $[n] = \{0, \ldots, n-1\}$.

[4] O nome da função, *StN*, vem de "string to number".