



Disciplina: Fundamentos da Computação		Visto:
Professor: Abrantes Araújo Silva Filho		
Aluno:		
Turma:	Semestre:	Valor: —
Data:	Diário 1: Fundamentos da Computação	

Unidade 1: Fundamentos da Computação
— Diário de Aprendizagem —

- Este **Diário de Aprendizagem** é uma das atividades integrantes da disciplina de **Fundamentos da Computação** do curso de Ciência da Computação, Universidade Vila Velha (UVV).
- A confecção do diário de aprendizagem é atividade **obrigatória e altamente recomendada** por três motivos: a) você aprenderá muito mais a matéria se mantiver o diário; b) ao entregar todos os diários ao professor você está cumprindo parte das atividades avaliativas que contam pontos na disciplina (10% da nota); e c) as provas bimestrais discursivas seguirão o formato e conteúdo das perguntas do diário.
- Se você tiver dificuldade em responder alguma questão do diário, estude novamente a matéria. Se você realmente entendeu a matéria, não deveria ter muita dificuldade para responder o diário.
- Responda com caneta ou lápis escuro (2B, 4B, 6B).
- Verifique no calendário de sua turma a **data de entrega**. Após uma rápida avaliação e visto pelo professor ou pelos monitores, seu diário será devolvido.
- O diário não será corrigido pelo professor: cabe a você estudar e dar a resposta correta para todas as questões. Obviamente o professor está à disposição para esclarecimento de dúvidas, e os monitores podem auxiliar caso você tenha dificuldade.
- Manter o diário de aprendizagem atualizado pode ser a diferença entre você aprender a matéria e ser aprovado, ou não aprender a matéria e não ser aprovado.
- Bons estudos!

1 Visão geral da unidade

1. Analise o **mapa mental** da “Unidade 1: Fundamentos da Computação”, no material da disciplina. Esse mapa mental detalhado mostra que os fundamentos da computação serão estudados em 5 grandes divisões (cada uma com subdivisões apropriadas). Quais são essas grandes divisões dos fundamentos da computação, de acordo com o mapa mental?

2 O que é ciência da computação?

2. Pesquise sobre o **Harold Abelson**, um cientista da computação americano e professor do MIT, que tem diversas contribuições importantes na computação e, principalmente, no ensino da ciência da computação. Escreva um pequeno texto descrevendo as principais contribuições de Abelson (certifique-se de incluir na sua resposta uma breve explicação sobre o livro: *Structure and Interpretation of Computer Programs* — SICP —, a obra prima de Abelson juntamente com **Gerald Jay Sussman**, outro cientista da computação do MIT). Dica: para aprender sobre o SICP, talvez você queira assistir um pedaço do vídeo “[Introdução à Série Pré-SICP](#)”¹.

3. Qual a **essência** da ciência da computação?

4. Por que a ciência da computação não é somente sobre programação nem somente sobre computadores?

¹<https://cmprz.me/presicp1>

5. Explique, com suas palavras, **o que é ciência da computação**.

6. Explique, de modo informal, o que é um **algoritmo**. Qual o nome do matemático persa que deu origem ao termo?

7. Ao resolver problemas, um cientista da computação sempre utiliza um **modelo geral da computação** para ajudar a pensar em tudo que é necessário. Faça um desenho ilustrando esse modelo e, a seguir, explique as três partes principais do modelo geral da computação.

8. Explique o que é o **conhecimento declarativo**, e demonstre que esse tipo de conhecimento, apesar de importante, não ajuda a resolver um problema computacional.

9. Explique o que é o **conhecimento imperativo**, e explique porque a ciência da computação está interessada nesse tipo de pensamento para a resolução de um problema computacional.

10. Calcule a $\sqrt{43}$ utilizando o **Método de Newton**. Demonstre, passo a passo, os cálculos efetuados. Pare quando você chegar ao resultado correto com cinco casas decimais (cinco números 0 após a vírgula).

11. Calcule o máximo divisor comum entre os números 192 e 56, utilizando o **Método de Euclides**. Demonstre, passo a passo, os cálculos efetuados.

12. Para o cálculo da $\sqrt{43}$ e do máximo divisor comum entre 192 e 56 você utilizou que tipo de conhecimento? Por quê?

3 Representação de dados

13. Pesquise sobre a **Arquitetura de von Neumann** e explique, brevemente, o que é e como funciona essa arquitetura.

14. Por que é importante encontrarmos uma **representação correta para o problema** que estamos tentando resolver?

15. Por que é importante encontrarmos uma **representação correta para a solução** que será apresentada como resposta para o problema que estamos tentando resolver?

16. O que é o **sistema unário** de numeração?

17. O que é o **sistema binário** de numeração?

18. O professor da CR6.100B tem um carro especial, com um hodômetro binário. Ao olhar tal hodômetro, você verificou que o carro já tinha percorrido 0000001010 km. Quantos quilômetros esse carro já andou (no sistema decimal)?

19. O que é um **bit** (b), um *binary digit*?

20. Em relação ao sistema binário, informe quantos números podemos representar, qual o menor e qual o maior, se tivermos um número binário com:

(a) 3 bits:

(b) 7 bits:

(c) 8 bits:

(d) 11 bits:

(e) 16 bits:

(f) 32 bits:

(g) 64 bits:

21. Por que os computadores “falam” binário e não decimal?

22. Qual componente eletrônico “fabrica” os 0 e 1 no computador? Quantos desses componentes existem hoje em dia em um processador?

23. Se eu tenho a necessidade de representar uma faixa numérica que vai do número 0 até o número 50, quantos transístores, no mínimo, serão necessários? Explique, em detalhes, o funcionamento desse componente eletrônico.

24. Quantos transístores, aproximadamente, existem hoje em um processador?

25. Explique a frase: “os bits 0 e 1 não existem de verdade, são apenas uma abstração”.

26. Em relação aos diferentes dispositivos que podem ser utilizados para criarmos uma abstração do sistema binário, explique como cada dispositivo abaixo representa os 0 e 1:

(a) Transístor:

(b) CD-ROM:

(c) Cartão perfurado:

(d) *Cores* de memória magnética:

(e) Disco rígido magnético:

(f) Fibra óptica:

27. Por que os dispositivos bi-estáveis podem ser utilizados para representar os 0 e 1 nos computadores?

28. O que é a **base** de um sistema numérico?

29. O que é o **valor posicional** de um algarismo?

30. Se o valor posicional de um algarismo qualquer em um número é dado por $n_i \times 10^i$, onde n representa o algarismo e i representa a posição desse algarismo (contada da direita para a esquerda, iniciando em zero), podemos afirmar que esse número está em qual sistema numérico? Por quê?

34. Quando estamos trabalhando com diferentes sistemas numéricos ao mesmo tempo, costumamos colocar uma indicação visual ao final do número para não confundirmos os sistemas. Por exemplo:

- O binário 101011 é representado por 101011_2 ou por $101011_{(2)}$
- O octal 640023 é representado por 640023_8 ou por $640023_{(8)}$
- O decimal 3510 é representado por 3510_{10} ou por $3510_{(10)}$
- O hexadecimal 35FA89 é representado por $35FA89_{16}$ ou por $35FA89_{(16)}$

Calcule o valor decimal dos seguintes números:

(a) $101011_{(2)}$

(b) $101011_{(8)}$

(c) $101011_{(10)}$

(d) $101011_{(16)}$

35. O que é um *nibble*?

36. O que é um *byte*?

37. Por que o *byte* (B) é uma unidade importante nos computadores?

38. Para armazenar a palavra “banana”, um computador precisa de:

(a) Quantos *bytes* (B)?

(b) Quanto *nibbles*?

(c) Quantos *bits* (b)?

39. Por que os prefixos e símbolos do sistema decimal são diferentes dos prefixos e símbolos do sistema binário? Por exemplo: por que no sistema decimal o prefixo de milhão é “mega” (M) e no sistema binário o prefixo de milhão é “mebi” (Mi)?

40. Se os computadores só entendem 0 e 1, como eles conseguem representar caracteres como letras e pontuação?

41. O que é um *encoding*?

42. O que é e para que serve o padrão ASCII?

43. Um programa de processamento de texto encontrou o seguinte padrão de bits:

01001111 01101100 11100001 00100001

O que esse programa exibiu na tela do computador?

44. Qual a diferença entre o ASCII original e o *Extended* ASCII?

45. Por que o padrão **Unicode** foi criado? Explique porque o Unicode é melhor do que o ASCII.

46. Em relação ao padrão Unicode, responda:

(a) O que é o Unicode?

(b) O que é um “Unicode Code Point”?

(c) Por que os “Unicode Code Points” não são um encoding como o ASCII?

47. Por que podemos afirmar que os *emojis* são texto?

48. Seu colega viu que o Code Point do emoji “*Face with Tears of Joy*”, ilustrado abaixo, é U+1F602.



Figura 1: *Face with Tears of Joy*

Seu colega também percebeu que o Code Point U+1F602 corresponde ao número decimal 128514. Ele transformou esse decimal em binário e encontrou o número 11111011000000010. Agora seu colega está afirmando que esse número binário é a representação correta desse emoji. Você sabe que ele está errado. Explique o motivo de seu colega estar errado.

49. Os três encodings mais utilizados para codificar os Code Points do Unicode são o UTF-8, UTF-16 e UTF-32. Explique o que é e quais as principais características de cada um desses encodings.

(a) UTF-8:

(b) UTF-16:

(c) UTF-32:

50. Qual encoding UTF é compatível com o encoding ASCII original de 7 bits?

51. O Unicode define a representação final dos caracteres, ou seja, o Unicode define como as letras serão apresentadas na tela do computador? Explique.

52. Explique como as cores são representadas no computador, utilizando-se o padrão RGB (*red-green-blue*).

53. Explique porque o padrão RGB consegue representar 16777216 cores diferentes.

54. No padrão RGB, que cores são essas?

(a) RGB(0, 0, 0):

(b) RGB(255, 0, 0):

(c) RGB(0, 255, 0):

(d) RGB(0, 0, 255):

(e) RGB(255, 255, 255):

(f) RGB(127, 127, 127):

(g) RGB(256, 256, 256):

55. O padrão RGB é um modelo **aditivo**. O que significa isso?

66. Se uma imagem tem profundidade de 1 bit, ela pode armazenar duas cores, preto e branco. Assim:
- (a) Se uma imagem tem profundidade de 8 bits, quantas cores ela pode suportar?
- _____
- (b) Se uma imagem tem profundidade de 16 bits, quantas cores ela pode suportar?
- _____
- (c) Se uma imagem tem profundidade de 24 bits, quantas cores ela pode suportar?
- _____

67. Assista ao vídeo abaixo no YouTube²:



Figura 3: *Let's enhance!*

Por que tornar uma imagem maior não a torna mais nítida?

68. O que são arquivos MIDI de áudio? Para que eles servem?

69. Cite dois formatos de arquivo de áudio, indicando as características desses arquivos.

²<https://cmprz.me/enhance1>

70. Em que situação você iria preferir um arquivo de áudio no formato MP3 ao invés de um arquivo de áudio no formato WAV?

71. Como um arquivo MIDI é diferente de um arquivo MP3 para uma mesma música?

72. Como é realizado o processo de gravação do som para sua representação no computador? Explique o processo, diferenciando a taxa de amostragem e a profundidade de bits.

73. O que é, basicamente, um arquivo de vídeo?

74. O que é o número de “quadros por segundo” (*frames per second* — FPS)?

75. Na compressão de vídeo são utilizadas duas estratégias ao mesmo tempo: intra-frame e inter-frames. Explique cada uma delas:

(a) Compressão intra-frame:

(b) Compressão inter-frames:

76. No contexto de armazenamento de vídeos, o que é um *key frame*?

77. Em relação aos formatos de arquivos de vídeo, diferencie os *containers* dos *codecs*.

78. Seu amigo disse que fez a “codificação de vídeo usando o formato DivX”. Seu amigo está certo ou errado? Por quê?

79. Explique como os vídeos são representados no computador.

80. Como o computador pode saber o que um determinado padrão de bits representa?

4 Representação de dados: anexos numéricos

81. Encontre o valor decimal dos números abaixo:

(a) $1110110011101_{(2)}$:

(b) $7243_{(8)}$:

(c) $CAFE_{(16)}$:

82. Transforme o número $35789_{(10)}$ em números nas seguintes bases numéricas:

(a) Em binário:

(b) Em octal:

(c) Em hexadecimal:

83. Transforme o número $111000111000_{(2)}$ em números nas seguintes bases numéricas:

(a) Em octal:

(b) Em hexadecimal:

84. Encontre o número hexadecimal correspondente aos seguintes números binários:

(a) $1111000010101010_{(2)}$:

(b) $1110001110001100_{(2)}$:

(c) $11111111111111111111111111111111_{(2)}$:

85. Encontre a representação binária dos seguintes números:

(a) $47301_{(8)}$:

(b) $BA00CA_{(16)}$:

86. Encontre o número octal que corresponde a $78AB2_{(16)}$:

87. Encontre o número hexadecimal que corresponde a $76543_{(8)}$:

88. Em binário, qual a diferença entre números **unsigned** e números **signed**?

89. Nos números **signed**, o que é o bit de sinal?

90. Qual a diferença do bit mais e menos significativo? Qual deles tem o maior valor posicional?

91. Considere que você tem um binário unsigned com 6 bits. Quantos números você consegue representar com esse número binário? Qual a faixa numérica desses números?

92. Faça a soma dos números 34 e 13, em binário unsigned.

93. Faça a multiplicação dos números 169 e 169, em binário unsigned.

94. Faça a subtração dos números 34 e 8, em binário unsigned.

95. Faça a divisão de 13 por 4, em binário unsigned, utilizando o método das subtrações sucessivas.

96. Faça a divisão de 210 por 8, em binário unsigned, utilizando o método do deslocamento.

97. Como é possível trabalhar com os números matemáticos, que são infinitos, em um computador que tem uma **memória finita**?

98. Explique como a memória de um computador pode ser entendida, e como ela funciona.

99. Cite 6 características das células de memória.

100. Se cada célula de memória armazena apenas 8 bits (1 byte), como a memória pode armazenar números de 32 bits?

101. O que é a **palavra** (*word*) de um computador (um processador)?

102. O que significa dizer que a arquitetura de um computador é de 64 bits?

103. O que é o **overflow**?

104. Considere que você está trabalhando em um programa e está usando variáveis inteiras com 8 bits, e apenas 8 bits. Demonstre, fazendo a soma binária unsigned, como ocorre overflow com a soma de 200 e 100.

105. Quais as 4 notações principais para armazenar binários inteiros negativos?

106. O que é a notação **sinal/magnitude**?

107. Represente o número $-356_{(10)}$ em binário de 16 bits em notação sinal/magnitude.

108. Se o binário $1111_{(2)}$ está em notação sinal/magnitude, que número decimal ele representa?

109. Quais as vantagens e desvantagens da notação sinal/magnitude?

110. O que é a notação **complemento de 1**?

111. Represente o número $-89_{(10)}$ em binário em notação de complemento de 1:

112. Se o binário $10110101_{(2)}$ está em notação complemento de 1, que número decimal ele representa?

113. Quais as vantagens e desvantagens da notação complemento de 1?

114. O que é a notação **complemento de 2**?

115. Represente o número $-235_{(10)}$ em notação de complemento de 2:

116. Se o binário $10110101_{(2)}$ está em notação complemento de 2, que número decimal ele representa?

117. Quais as vantagens e desvantagens da notação complemento de 2?

118. Considere que você está trabalhando com números binários de 4 bits, em notação complemento de 2. Demonstre que a soma dos números 4 e -3 , em complemento de 2, funciona e não causa overflow.

119. Quais as duas “formas rápidas” para encontrarmos o complemento de 2 de um número binário?

120. Faça a soma dos números 4 e -6 , considerando que você está trabalhando com números binários de 4 bits em complemento de 2. Demonstre que não ocorre overflow.

121. Na aritmética binária com números em complemento de 2, em que situações pode ocorrer overflow? Como podemos detectar se ocorreu overflow?

122. Demonstre que a soma de 4 e 5, considerando que você está trabalhando com números binários de 4 bits em complemento de 2, causa overflow.

123. Demonstre que a soma de -4 e -6 , considerando que você está trabalhando com números binários de 4 bits em complemento de 2, causa overflow.

124. O que é a notação **enviesada** (também chamada de notação de **excesso**)?

125. Represente o número 5 em binário de 4 bits na notação enviesada, com viés de 8:

126. Se você está trabalhando com números binários com 6 bits, em notação enviesada com viés de 64, como é a representação binária do número zero?

127. Quais as vantagens e desvantagens da notação enviesada?

128. Na notação enviesada, se um número começa com o bit “0”, esse número é positivo ou negativo?

129. Na notação enviesada, um número binário que só tem bits “0” é o:

- A. O zero
- B. O um
- C. O número mais positivo
- D. O número mais negativo
- E. Nenhuma das respostas anteriores

130. Qual o valor decimal do binário $1001_{(2)}$, se ele estiver:

(a) Em notação unsigned:

(b) Em notação sinal/magnitude:

(c) Em notação complemento de 1:

(d) Em notação complemento de 2:

(e) Em notação enviesada com viés de 8:

131. O que é a notação de **ponto fixo**?

132. Considere que o número $0110.110_{(2)}$ é um binário unsigned em notação de ponto fixo. Que número decimal ele representa?

133. Encontre o valor decimal do seguinte binário unsigned em ponto fixo: $0000.0110_{(2)}$:

134. Encontre a representação binária unsigned, em ponto fixo, dos seguintes números:

(a) $6,375_{(10)}$:

(b) $27,09375_{(10)}$:

135. O computador consegue representar todos os números fracionários, de modo exato, em notação de ponto fixo? Por quê?

136. Quando estamos trabalhando com números em notação de ponto fixo, o que significam as notações:

(a) Up.f:

(b) Qp.f:

137. Represente o número $-2,375_{(10)}$ em notação Q4.4.

138. O que é a **resolução** em uma notação de ponto fixo? O que ela indica?

139. O que é o **underflow**?

140. O que é a notação de **ponto flutuante**?

141. O que é a notação científica? Quando podemos dizer que ela está na forma normalizada?

142. O que é o **significando** de um número em notação científica?

143. Por que o ponto decimal, na notação de ponto flutuante, “flutua”?

144. Como é possível saber se um número de ponto flutuante, na base 2, está na forma normalizada?

145. O que é o padrão IEEE-754?

146. O expoente, no padrão IEEE-754, pode ter três funções diferentes. Que funções são essas?

147. Quando o padrão IEEE-754 diz que a precisão do formato “*Single Precision*” é de 24 bits de precisão, o que isso quer dizer?

148. Represente o número $228,0_{(10)}$ em ponto flutuante “*Half Precision*”:

149. Que “macete” o padrão IEEE-754 utiliza para ganhar 1 bit de precisão na representação do significando?

150. No padrão IEEE-754, que tipo de notação é utilizada para representar o expoente? Qual o viés utilizado?

151. No padrão IEEE-754, há algum uso para a notação sinal/magnitude?

152. Represente o número $-0,75_{(10)}$ em ponto flutuante “*Half Precision*”:

153. A notação de ponto flutuante do padrão IEEE-754 consegue representar todos os números reais? Sim ou não? Por quê?

154. No padrão IEEE-754, o que um expoente que é formato totalmente por bits “1” indica?

155. Faça uma tabela que mostre os limites e as faixas de representação dos números de ponto flutuante nos formatos *Half*, *Single*, *Double* e *Quadruple precision*. Você pode considerar apenas a faixa positiva.

156. Quando ocorre underflow na notação de ponto flutuante?

157. Um grande problema da notação de ponto flutuante é que os cálculos são aproximados e diversos arredondamentos são feitos. Outro problema é que algumas propriedades matemáticas da adição e da multiplicação também não valem em ponto flutuante. Quais propriedades matemáticas não valem na:

(a) Adição em ponto flutuante:

(b) Multiplicação em ponto flutuante:

158. Qual o método padrão de arredondamento definido pelo padrão IEEE-754?

159. Cite um exemplo clássico de um problema real causado pela imprecisão da notação de ponto flutuante.

160. O que é a notação BCD?

5 Algoritmos

161. Já sabemos que a Ciência da Computação é a ciência que projeta e implementa algoritmos para a resolução de problemas. Em relação a essa definição, responda:

(a) Explique qual o interesse do cientista da computação quando ele **projeta** algoritmos?

(b) Explique quais os três interesses do cientista da computação quando ele **implementa** algoritmos?

(c) Explique qual o interesse do cientista da computação quando ele **resolve problemas** através de algoritmos?

162. A área da computação pode ser estudada em diferentes cursos, tais como: Ciência da Computação, Engenharia da Computação, Sistemas de Informação, Engenharia de Software e outros. O que diferencia esses cursos entre si?

163. Cite 3 disciplinas que preparam o cientista da computação para o estudo das propriedades formais dos algoritmos.

164. Cite 3 disciplinas que preparam o cientista da computação para o estudo das propriedades matemáticas dos algoritmos.

165. Cite 3 disciplinas que preparam o cientista da computação para o estudo da realização linguística dos algoritmos.

166. Cite 3 disciplinas que preparam o cientista da computação para o estudo da realização em hardware dos algoritmos.

167. Cite 3 disciplinas que preparam o cientista da computação para o estudo da realização virtual dos algoritmos.

168. Cite 3 disciplinas que preparam o cientista da computação para a criação de aplicações dos algoritmos.

169. Por que a ciência da computação não é somente sobre programação nem somente sobre computadores?

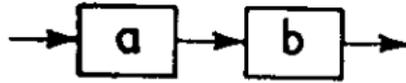
170. Escreva uma definição **informal** para o conceito de algoritmo. Explique a definição.

171. Qualquer algoritmo, desde os mais simples até os mais complexos e sofisticados, pode ser criado utilizando-se apenas esses 4 conceitos. Cite quais são esses conceitos.

- (a) _____
- (b) _____
- (c) _____
- (d) _____

172. O que são **operações seqüenciais**?

173. Que tipo de operação está representado no fluxograma abaixo?



174. Em relação às operações seqüenciais em um algoritmo, explique o que são as **sentenças** (*statements*).

175. Em um algoritmo, cada **sentença** (*statement*) deve ser específica, única, simples, atômica, primitiva. O que isso quer dizer?

176. Por que as **sentenças** devem ser atômicas?

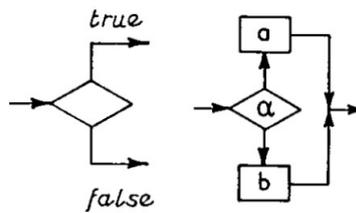
177. As **operações condicionais** têm dois sinônimos consagrados. Quais são esses sinônimos?

178. O que são as **operações condicionais**?

179. O que é uma **expressão booleana**?

180. Qual a relação existente entre as operações condicionais e as expressões booleanas?

181. A figura abaixo representa operações condicionais através de um fluxograma. O que a estrutura marcada com a letra grega α representa? Por que essa estrutura é importante?



182. O que são as **operações de repetição**?

183. Existem, a grosso modo, dois grandes tipos de operações de repetição: as operações onde uma determinada ação pode ser **repetida 0 ou mais vezes** e as operações onde uma determinada ação pode ser **repetida 1 ou mais vezes**. Isso significa que existem instruções de repetição que não garantem que a ação desejada será executada nem uma única vez, e existem instruções de repetição que garantem que a ação desejada será executada, pelo menos, uma vez.

(a) Desenhe um fluxograma que represente uma ação de repetição que garante que uma ação desejada seja executada pelo menos uma vez:

(b) Desenhe um fluxograma que represente uma ação de repetição que não garante que uma ação desejada seja executada pelo menos uma vez:

184. O que são **estruturas de dados**?

185. Qual a estrutura de dados mais simples?

186. Por que consideramos que:

Programas = Algoritmos + Estruturas de Dados

187. Por que os algoritmos devem ser expressos em um nível extremamente alto de detalhes?

188. O que queremos dizer falamos que **um algoritmo resolve todos os problemas de uma mesma “classe”**?

189. Em relação à criação de algoritmos para a solução de problemas, sabe-se que existem **problemas insolúveis**. O que são esses problemas? Que matemático provou que existem problemas insolúveis?

190. Em relação à criação de algoritmos para a solução de problemas, a Conjectura de Goldbach, “Todo inteiro par positivo maior do que 2 é a soma de dois números primos”, é um exemplo de um problema para o qual nós não sabemos se existe ou não uma solução algorítmica. O que isso quer dizer?

191. O problema do “Caixeiro Viajante” é um dos problemas clássicos para os quais não existe uma solução algorítmica eficiente. Pesquise sobre esse problema e explique o que é e porque não tem solução eficiente.

192. Qual a diferença entre um **algoritmo** e uma **heurística**?

193. A definição formal de algoritmo que adotaremos neste curso é a seguinte:

Definição de Algoritmo

É uma coleção bem ordenada de operações efetivamente computáveis, definidas e não ambíguas que, quando executada sobre uma entrada produz uma saída e termina em uma quantidade finita de passos e de tempo

Em relação a essa definição forma, responda:

(a) O que significa “coleção bem ordenada”?

(b) O que significa “operações efetivamente computáveis”?

(c) O que significa “definidas e não ambíguas”?

(d) O que significa “sobre uma entrada”?

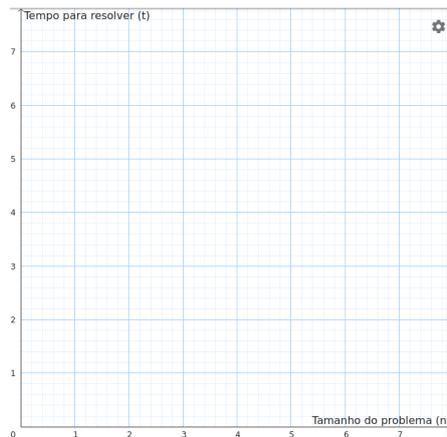
(e) O que significa “produz uma saída”?

(f) O que significa dizer que o algoritmo “termina”?

(g) O que significa dizer que o algoritmo termina “em uma quantidade finita de passos”?

(h) O que significa dizer que o algoritmo termina “em uma quantidade finita de tempo”?

194. Quando medimos a eficiência de um algoritmo geralmente estamos interessados no tempo t que esse algoritmo leva para resolver um problema de tamanho n , conforme o gráfico abaixo:



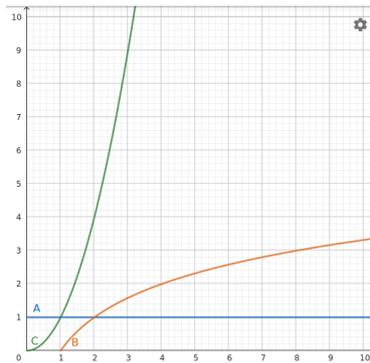
A particularidade dessa medida de eficiência dos algoritmos é que o tempo t não é medido em número de segundos ou minutos, seja, não é medido em tempo de “relógio”, mas, sim, como o **número de operações que o algoritmo realiza**. Explique porque o tempo de execução é medido em número de operações realizadas e não pelo tempo de relógio.

195. O que é a notação “Big-O”? Para que ela serve?

196. Quando falamos que um algoritmo é $O(n)$, o que estamos querendo dizer?

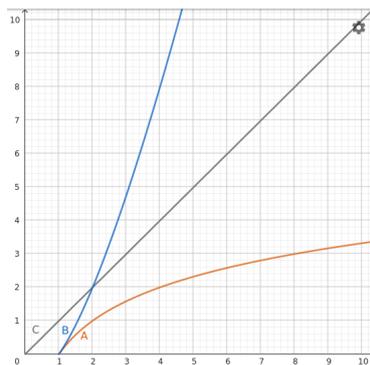
197. Quando falamos que um algoritmo é $O(\log_2 n)$, o que estamos querendo dizer?

198. A figura abaixo mostra o tempo de execução de três algoritmos diferentes para resolver o mesmo problema:



Sabe-se que esses algoritmos tem tempos de $O(1)$, $O(n^2)$ e $O(\log_2 n)$. Qual dos algoritmos acima (A, B ou C) é o algoritmo de tempo quadrático?

199. A figura abaixo mostra o tempo de execução de três algoritmos diferentes para resolver o mesmo problema:



Sabe-se que esses algoritmos tem tempos de $O(n)$, $O(n \log_2 n)$ e $O(\log_2 n)$. Qual dos algoritmos acima (A, B ou C) é o algoritmo de tempo log-linear?

200. Que algoritmo é mais lento: um de tempo $O(n \log_2 n)$ ou um de tempo $O(2^n)$?

201. Por que a linguagem natural não é um bom meio de expressar um algoritmo?

202. Por que linguagens de programação não são um bom meio de expressar um algoritmo?

203. Por que fluxogramas são um bom meio de expressar um algoritmo?

204. O que é **pseudocódigo**?

205. Por que pseudocódigo é um bom meio de expressar um algoritmo?

206. Por que não podemos começar a solucionar um problema programando, ou seja, colocando a mão no teclado?

6 Pensamento Computacional

207. Cite três ou mais características presentes em **problemas complexos**:

208. O que é o **pensamento computacional**?

209. Para que serve o pensamento computacional?

210. Quais os cinco componentes principais do pensamento computacional:

- (a) _____
- (b) _____
- (c) _____
- (d) _____
- (e) _____

211. O que é a **decomposição**?

212. Quando fazemos a decomposição de um problema, estamos interessados em hierarquizar, modularizar e manter a regularidade entre as partes decompostas. No contexto da decomposição, o que quer dizer:

(a) Hierarquia:

(b) Modularidade:

(c) Regularidade:

213. O que é o **reconhecimento de padrões**?

214. O que é, de modo geral, a **abstração**?

215. Por que a **representação de dados** é essencial no pensamento computacional?

216. O que é o **pensamento algorítmico**?

7 Abstração

217. A nossa capacidade de **abstração** é a chave para a resolução de problemas complexos na computação. Em geral a abstração tem 2 grandes objetivos: a remoção de detalhes/foco no essencial, e a capacidade de generalização. Explique o que quer dizer cada uma dessas coisas:

(a) Remoção de detalhes/foco no essencial:

(b) Generalização

218. O que é a **interface** fornecida por uma abstração?

219. Uma interface separa o uso da implementação. Dê um exemplo de uma abstração cuja interface permaneceu a mesma, mesmo que a implementação interna tenha mudado. Explique como isso ocorreu.

220. Por que a abstração facilita a decomposição?

221. Dê três exemplos de abstração que você utiliza no dia a dia. Indique qual é a remoção de detalhe/foco no essencial, e a generalização de cada abstração.

222. Por que funções são um ótimo exemplo de abstração?

223. Por que a **interface** é uma barreira entre o uso e a implementação de uma abstração?

224. Por que a **interface** é um “contrato” entre quem usa e quem implementa uma abstração?

225. Qual o componente mais importante do pensamento computacional para o controle da complexidade de problemas?
