

Lista com 100 exercícios

1. Fibonacci Recursivo: Escreva um método em Java para calcular o n-ésimo termo da sequência de Fibonacci de forma recursiva.
2. Verificador de Palíndromo: Crie um método em Java que verifique se uma string é um palíndromo.
3. Ordenação por Seleção: Implemente o algoritmo de ordenação por seleção em Java.
4. Verificação de Anagramas: Crie um método em Java que verifique se duas strings são anagramas uma da outra.
5. Contagem de Palavras em um Texto: Escreva um programa em Java que conte o número de palavras em um texto.
6. Validação de CPF: Desenvolva uma função em Java para validar se um número de CPF é válido ou não.
7. Soma de Matrizes: Implemente um método em Java para somar duas matrizes.
8. Contagem de Vogais e Consoantes: Crie um programa em Java que conte o número de vogais e consoantes em uma string.
9. Conversão de Temperatura: Escreva um método em Java para converter uma temperatura de Celsius para Fahrenheit.
10. Maior Subsequência Crescente: Implemente um algoritmo em Java para encontrar a maior subsequência crescente em uma sequência de números.

11. Busca Binária: Desenvolva uma função em Java para realizar busca binária em um array ordenado.
12. Validação de Expressões Matemáticas: Crie um programa em Java para validar se uma expressão matemática está bem formada.
13. Compressão de Strings: Escreva um método em Java para comprimir uma string substituindo repetições de caracteres por contagem.
14. Implementação de Lista Encadeada: Implemente uma lista encadeada em Java com métodos para inserção, remoção e busca.
15. Conversão de Números Romanos: Desenvolva uma função em Java para converter números romanos para inteiros.
16. Geração de Números Primos: Crie um programa em Java para gerar os n primeiros números primos.
17. Ordenação por Inserção: Implemente o algoritmo de ordenação por inserção em Java.
18. Validação de Email: Escreva uma função em Java para validar se uma string representa um email válido.
19. Busca em Profundidade (DFS): Implemente o algoritmo de busca em profundidade (DFS) em um grafo representado por uma matriz de adjacência.
20. Verificação de Balanceamento de Parênteses: Desenvolva um programa em Java para verificar se uma expressão matemática tem parênteses balanceados.

21. Esses desafios abrangem uma variedade de conceitos e dificuldades, proporcionando uma oportunidade para os programadores praticarem e aprimorarem suas habilidades em Java.
22. Média de uma Matriz: Escreva um método em Java para calcular a média dos elementos de uma matriz.
23. Maior Subsequência Comum: Implemente um algoritmo em Java para encontrar a maior subsequência comum entre duas strings.
24. Verificação de Primalidade: Desenvolva uma função em Java para verificar se um número é primo ou não.
25. Cálculo de Fatorial: Crie um método em Java para calcular o fatorial de um número.
26. Troco de Moedas: Escreva um programa em Java que determine a quantidade mínima de moedas para representar um determinado valor em centavos.
27. Ordenação por MergeSort: Implemente o algoritmo de ordenação MergeSort em Java.
28. Decodificação de Huffman: Desenvolva um programa em Java para decodificar uma sequência codificada usando o algoritmo de Huffman.
29. Contagem de Inversões em uma Lista: Escreva um método em Java para contar o número de inversões em uma lista de números.
30. Encontrar o Elemento Único: Implemente um método em Java para encontrar o único elemento não repetido em um array de inteiros.
31. Subconjuntos de um Conjunto: Crie um programa em Java para gerar todos os subconjuntos de um conjunto dado.

32. Soma Máxima de Subconjunto: Desenvolva um algoritmo em Java para encontrar a soma máxima de um subconjunto contíguo em um array de inteiros.
33. Validação de Senha: Escreva uma função em Java para validar se uma senha atende aos critérios de segurança, como ter pelo menos 8 caracteres, pelo menos uma letra maiúscula, uma letra minúscula e um dígito.
34. Implementação de Árvore Binária: Implemente uma árvore binária em Java com métodos para inserção, remoção e busca.
35. Verificação de Ciclo em um Grafo: Crie um programa em Java para verificar se um grafo direcionado contém um ciclo.
36. Contagem de Ocorrências em uma String: Escreva um método em Java para contar o número de ocorrências de uma sub-string em uma string dada.
37. Cálculo de Potência: Desenvolva uma função em Java para calcular o resultado de um número elevado a uma potência inteira.
38. Busca em Largura (BFS): Implemente o algoritmo de busca em largura (BFS) em um grafo representado por uma matriz de adjacência.
39. Ordenação por QuickSort: Implemente o algoritmo de ordenação QuickSort em Java.
40. Validação de Data: Escreva uma função em Java para validar se uma data fornecida está no formato correto e é uma data válida.
41. Verificação de Subsequência: Desenvolva um método em Java para verificar se uma string é uma subsequência de outra string.

42. Verificação de Pangrama: Crie um programa em Java para verificar se uma frase é um pangrama, ou seja, se contém todas as letras do alfabeto pelo menos uma vez.
43. Conversão de Números Binários: Implemente um método em Java para converter um número binário em sua representação decimal.
44. Soma de Dois Números em um Array: Escreva um programa em Java que determine se existe algum par de números distintos em um array cuja soma seja igual a um valor dado.
45. Implementação de Pilha: Implemente uma pilha em Java com métodos push, pop e peek.
46. Contagem de Inversões em uma Matriz: Desenvolva um método em Java para contar o número de inversões necessárias para transformar uma matriz em outra matriz dada.
47. Validação de Sudoku: Crie um programa em Java para validar se uma matriz representa um tabuleiro de Sudoku válido.
48. Encontrar o Número Desaparecido: Implemente um método em Java para encontrar o número que está ausente em uma sequência de números de 1 a n, onde apenas um número está ausente.
49. Substituição de Espaços em uma String: Escreva um método em Java para substituir todos os espaços em branco em uma string por "%20".
50. Conversão de Horas para Minutos: Desenvolva uma função em Java para converter um determinado número de horas em minutos.
51. Verificação de Sequência Fibonacci: Crie um programa em Java para verificar se uma sequência de números é uma sequência de Fibonacci.

52. Cálculo de Média Ponderada: Implemente um método em Java para calcular a média ponderada de um conjunto de números e pesos dados.
53. Implementação de Fila: Implemente uma fila em Java com métodos enqueue, dequeue e peek.
54. Verificação de Submatriz Identidade: Escreva um programa em Java para verificar se uma matriz quadrada é uma submatriz identidade em outra matriz maior.
55. Ordenação por RadixSort: Implemente o algoritmo de ordenação RadixSort em Java.
56. Encontrar a Raiz Quadrada: Desenvolva uma função em Java para calcular a raiz quadrada de um número sem usar as funções de raiz quadrada padrão.
57. Validação de IP: Crie um programa em Java para validar se uma string representa um endereço IP válido.
58. Contagem de Sequência de Fibonacci: Escreva um método em Java para contar o número de dígitos em um determinado termo da sequência de Fibonacci.
59. Verificação de String Rotacionada: Implemente uma função em Java para verificar se uma string é uma rotação circular de outra string.
60. Cálculo de Média Móvel: Desenvolva um programa em Java para calcular a média móvel de uma sequência de números.
61. Implementação de Heap: Implemente uma estrutura de heap em Java com métodos para inserção e remoção.

62. Conversão de String para Inteiro: Escreva um método em Java para converter uma string que representa um número em um inteiro.
63. Verificação de Palavras Cruzadas Válidas: Crie um programa em Java para verificar se uma matriz de letras forma palavras cruzadas válidas.
64. Contagem de Caminhos em um Grafo: Implemente um método em Java para contar o número de caminhos possíveis entre dois vértices em um grafo direcionado acíclico (DAG).
65. Ordenação por CountingSort: Implemente o algoritmo de ordenação CountingSort em Java.
66. Decodificação de Base64: Desenvolva um programa em Java para decodificar uma string codificada em Base64.
67. Verificação de Permutação: Escreva um método em Java para verificar se uma string é uma permutação de outra string.
68. Cálculo de Área de um Polígono: Implemente uma função em Java para calcular a área de um polígono dado seus vértices.
69. Validação de Expressão Regular: Crie um programa em Java para validar se uma string corresponde a uma expressão regular dada.
70. Busca de Palavra em Matriz de Letras: Desenvolva um método em Java para encontrar todas as ocorrências de uma palavra em uma matriz de letras.
71. Cálculo de Fibonacci com Matriz: Escreva um método em Java para calcular o n-ésimo termo da sequência de Fibonacci usando matriz de transformação.

72. Validação de Cartão de Crédito: Implemente uma função em Java para validar se um número de cartão de crédito é válido usando o algoritmo de Luhn.
73. Contagem de Subconjuntos que Soma 0: Crie um programa em Java para contar o número de subconjuntos de uma matriz que somam zero.
74. Ordenação por ShellSort: Implemente o algoritmo de ordenação ShellSort em Java.
75. Cálculo de Números Perfeitos: Desenvolva um método em Java para encontrar todos os números perfeitos menores que um número dado.
76. Contagem de Instruções em um Programa: Escreva um programa em Java para contar o número total de instruções em um programa Java dado.
77. Verificação de Conexidade em um Grafo: Implemente um programa em Java para verificar se um grafo não direcionado é conexo ou não.
78. Cálculo de Mínimo Múltiplo Comum (MMC): Desenvolva uma função em Java para calcular o mínimo múltiplo comum (MMC) de dois números.
79. Implementação de Árvore AVL: Implemente uma árvore AVL em Java com métodos para inserção, remoção e busca.
80. Verificação de Sequência Aritmética: Escreva um método em Java para verificar se uma sequência de números é uma sequência aritmética.
81. Decodificação de URL: Crie um programa em Java para decodificar uma string codificada em formato de URL.
82. Contagem de Quadrados em uma Matriz: Implemente um método em Java para contar o número de quadrados de tamanho k em uma matriz.

83. Ordenação por BucketSort: Implemente o algoritmo de ordenação BucketSort em Java.
84. Cálculo de Médiana: Desenvolva uma função em Java para calcular a mediana de um conjunto de números.
85. Validação de Expressão Matemática: Escreva um programa em Java para validar se uma expressão matemática está correta e bem formada.
86. Busca de Palavra em Dicionário: Crie um método em Java para buscar uma palavra em um dicionário usando pesquisa binária.
87. Cálculo de Coeficiente Binomial: Implemente uma função em Java para calcular o coeficiente binomial de dois números inteiros.
88. Verificação de Sequência Geométrica: Escreva um método em Java para verificar se uma sequência de números é uma sequência geométrica.
89. Cálculo de Distância entre Pontos: Desenvolva uma função em Java para calcular a distância entre dois pontos em um plano cartesiano.
90. Implementação de Árvore B: Implemente uma árvore B em Java com métodos para inserção, remoção e busca.
91. Verificação de Equilíbrio de Parênteses: Crie um programa em Java para verificar se uma expressão matemática tem parênteses, colchetes e chaves balanceados.
92. Contagem de Quadrados em uma Sequência: Escreva um método em Java para contar o número de quadrados perfeitos em uma sequência de números.

93. Ordenação por CocktailSort: Implemente o algoritmo de ordenação CocktailSort em Java.
94. Cálculo de Distância Levenshtein: Desenvolva uma função em Java para calcular a distância de edição entre duas strings usando o algoritmo de distância de Levenshtein.
95. Validação de Expressão Regular (Regex): Crie um programa em Java para validar se uma string corresponde a uma expressão regular fornecida como entrada.
96. Busca de Palavra em Matriz de Letras (DFS): Implemente um método em Java para encontrar todas as ocorrências de uma palavra em uma matriz de letras usando busca em profundidade (DFS).
97. Cálculo de Fibonacci com Complexidade Logarítmica: Escreva um método em Java para calcular o n-ésimo termo da sequência de Fibonacci com complexidade logarítmica.
98. Verificação de Sequência de Collatz: Desenvolva um programa em Java para verificar se uma sequência de números segue a conjectura de Collatz.
99. Decodificação de Morse: Implemente um programa em Java para decodificar uma string codificada em código Morse.
100. Contagem de Inversões em uma Árvore: Escreva um método em Java para contar o número de inversões em uma árvore binária de pesquisa.
101. Implementação de Grafo Ponderado: Implemente um grafo ponderado em Java com métodos para adicionar vértices, adicionar arestas e calcular o menor caminho entre dois vértices.