Material de Apoio

- Algoritmos matemáticos (Fatorial, Fibonacci, Números Primos, etc);
- Algoritmos de Ordenação;
- Algoritmos de Grafo;
- Algoritmos de Árvore;
- Referências do C++ e Python;
- Códigos com conversões (bin/dec, int/string, string/int);
- Referências de String (Códigos de exemplo, etc.);

Referência do Livro

Índice

SAT (problema de satisfatibilidade booleana) é o problema de atribuir valores booleanos a variáveis para satisfazer uma determinada fórmula booleana. A fórmula booleana geralmente será dada em CNF (forma normal conjuntiva), que é uma conjunção de múltiplas cláusulas, onde cada cláusula é uma disjunção de literais (variáveis ou negação de variáveis). 2-SAT (2-satisfatibilidade) é uma restrição do problema SAT, em 2-SAT cada cláusula possui exatamente dois literais. Aqui está um exemplo de problema 2-SAT.

O algoritmo Berlekamp-Massey é um algoritmo que encontrará o menor registrador de deslocamento de feedback linear (LFSR) para uma determinada sequência de saída binária. O algoritmo também encontrará o polinômio mínimo de uma sequência linearmente recorrente em um corpo arbitrário . O requisito de campo significa que o algoritmo Berlekamp – Massey exige que todos os elementos diferentes de zero tenham um inverso multiplicativo. [1] Reeds e Sloane oferecem uma extensão para manusear um anel . [2]

1.7 Crivo de Eratosthenes 8

O Crivo de Eratóstenes é um algoritmo e um método simples e prático para encontrar números primos até um certo valor limite.

1.8 Deteccao de ciclo - Tortoise and Hare 9

O Algoritmo de Floyd, criado por Robert Floyd na década de 1970, foi desenvolvido originalmente para encontrar ciclos em listas ligadas, mas esse princípio pode ser utilizado para encontrar valores inteiros duplicados em um array se entendermos esse array específico como um tipo de lista ligada. Isso só é possível quando o array está no intervalo [1..n] e a lista tem tamanho n+1. O algoritmo tem as seguintes pré-condições:

1.9 Division Trick 9

Um algoritmo de divisão é um algoritmo que, dados dois inteiros N e D (respectivamente o numerador e o denominador), calcula seu quociente e/ou resto , o resultado da divisão euclidiana . Alguns são aplicados manualmente, enquanto outros são empregados por projetos de circuitos digitais e software.

Os algoritmos de divisão se enquadram em duas categorias principais: divisão lenta e divisão rápida. Algoritmos de divisão lenta produzem um dígito do quociente final por iteração. Exemplos de divisão lenta incluem restauração , restauração sem desempenho, não restauração e divisão SRT . Os métodos de divisão rápida começam com uma grande aproximação do quociente final e produzem o dobro de dígitos do quociente final em cada iteração. [1] Os algoritmos Newton-Raphson e Goldschmidt se enquadram nesta categoria.

Variantes destes algoritmos permitem usar algoritmos de multiplicação rápida . O resultado é que, para números inteiros grandes, o tempo de computador necessário para uma divisão é o mesmo, até um fator constante, que o tempo necessário para uma multiplicação, qualquer que seja o algoritmo de multiplicação usado.

1.10 Equacao Diofantina Linear 10

Teorema: Seja x0 e y0 uma solução particular, arbitrariamente dada, da equação ax+by=c, onde mdc(a,b)=1. Então as soluções da equação são da forma x=x0+bt e y=y0-at, para t variando nos inteiros.

1.11 Euclides estendido 10

O Algoritmo de Euclides estendido é uma extensão do algoritmo de Euclides, que, além de calcular o máximo divisor comum (MDC) entre a, b.

1.12 Exponenciacao rapida 10

1.13 Fast Walsh Hadamard Transform 10

Fast Walsh Hadamard Transform , é um algoritmo eficiente ordenado de Hadamard para calcular a transformada de Walsh Hadamard (WHT). O cálculo normal do WHT tem complexidade N = 2 m , mas o uso do FWHT reduz o cálculo para O(n 2) . O FWHT requer O(n logn) operações de adição e subtração. É um algoritmo de dividir e conquistar que divide o WHT recursivamente.

$$H_0 = 1$$

$$H_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$H_2 = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

$$H_3 = \frac{1}{2^{3/2}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}$$

1.14 FFT 11

Uma transformada rápida de Fourier (FFT) é um algoritmo que calcula a transformada discreta de Fourier (DFT) de uma sequência, ou seu inverso (IDFT). A análise de Fourier converte um sinal de seu domínio original (geralmente tempo ou espaço) em uma representação no domínio da frequência e vice-versa. A DFT é obtida decompondo uma sequência de valores em componentes de diferentes frequências. [1] Esta operação é útil em muitos campos, mas calculá-la diretamente a partir da definição costuma ser muito lento para ser prático. Uma FFT calcula rapidamente essas transformações fatorando a matriz DFT em um produto de fatores esparsos (principalmente zero).