

1 -

a - A sequência de fibonacci tem pior caso de O(2^n) para o caso recursivo e O(n) para o caso iterativo, melhor caso 2^n/2 para o caso recursivo e n/2 para o caso iterativo, caso médio estaria entre esses dois caso nas duas versões. A melhor versão seria a versão iterativa.

versão iterativa: AALista2FibonacciNRecursivo

versão recursiva:AALista2Finacci

b - O algoritmo de permutação tem pior caso de O(n\*n!) para o caso recursivo e O(n^2) para o caso iterativo, o melhor caso seria constante.

versão iterativa: AALista2Permutacoes

versão recursiva:AALista2PermutacoesNonRecursive

2 -

a - Esse trecho de código apresenta um for que vai de 2 até um número antes ou igual a n, então o custo será n - 3, pior caso será O(n).

b- T(n) = 2 T(n/2) + n  
 = 2 [2 T(n/4) + n/2] + n  
 = 4 T(n/4) + 2n  
 = 4 [2 T(n/8) + n/4] + 2n  
 = 8 T(n/8) + 3n

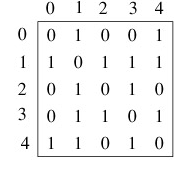
= 2k T(n/2k) + k n <¬ custo da função

= 2k T(n/2k) + k n  
 = 2log2 n T(1) + (log2n) n  
 = n + n log2 n   
 = O(n log n) <¬ complexidade de tempo

3 - Defina e apresente exemplos de matriz de incidência, matriz de adjacência e lista de

adjacência. Adicionalmente, descreva o impacto (vantagens e desvantagens) da utilização de matriz de adjacência e lista de adjacência.

Matriz de adjacência: é um array de duas dimensões de tamanho V x V, onde V é o número de vértices no grafo. Por exemplo, no array adj[][], uma célula adj[i][j] = 1 indica que a célula na linha i e coluna j tem valor igual a 1 e indica que tem um aresta entre os vértices i e j. Matriz de adjacência para grafos não direcionados é sempre simétrica. A matriz de adjacência é também usada para representar grafos com peso. se adj[i][j] = w, então existe uma aresta do vértice i para o vértice j com peso w.



*Representação de um grafo através de uma matriz de adjacência*

Vantagens da matriz de adjacência: Representação é mais de implementar e seguir. Remover uma aresta demora tempo de O(1). Consultas como se tem um aresta do vértice ‘u’ para o vértice ‘v’ são eficientes e podem ser feitas em O(1).

Desvantagens da matriz de adjacência: Consome muito espaço O(V^2). Mesmo se o grafo é esparso( contém menos números de arestas), ele consome o mesmo espaço. adicionar um vértice demora tempo de O(V^2).

Lista de Adjacência: É um array de listas ligadas. O tamanho do array é igual ao número de vértices. O array array[]. Uma entrada array[i] representa a lista ligada de vértices adjacentes até o vértice i. Essa representação pode também ser usada para representar um grafo com pesos. Os pesos das arestas podem ser guardados em nós de listas ligadas.

*Programa da lista de adjacência: AALista2GrafoListaAdj*

Vantagens da lista de adjacência: Salva o espaço O(|V| + |E|). No pior caso, podem haver C(V,2) de arestas em um grafo, portanto consumindo O(V^2) de espaço.Adicionar um vértice é mais fácil do que na matriz de adjacência.

Desvantagens: Consultas como se existe uma aresta do vértice u para o vértice v não são eficientes e podem ser feitas em um tempo de O(V).

Matriz de incidência: matriz de incidência é a matriz que representa um grafo tal que, com a ajuda da matriz pode-se desenhar o grafo. Essa matriz pode ser denotada como [Ac], como em toda matriz, também existe linhas e colunas em uma matriz de incidência [Ac].

As linhas da matriz [Ac] representam o número de nós e as colunas da matriz [Ac] representam os ramos do grafo. Se existem ‘n’ números de linhas em uma matriz de incidência, isso significa que em um grafo existem ‘n’ nós, igualmente se existem m colunas numa matriz de incidência, isso significa que naquele grafo existem m ramos.

5 - Defina, explicando as principais características e exemplifique:

(A) Enumeração explícita x implícita.

Na enumeração explícita, o algoritmo gera consecutivamente a árvore de enumeração com todas as possíveis soluções e memoriza a melhor solução corrente do problema examinado.

Na enumeração implícita, o algoritmo corta consecutivamente de todos os subconjuntos, o espaço da solução em que, sob condições pré-determinadas, nenhuma solução ótima poderia ser esperada. O algoritmo gera consecutivamente a árvore de enumeração com todos os espaços restantes.

(B) Programação Dinâmica.

Programação dinâmica é uma técnica de programação que é comumente usada em uma fórmula recorrente e um ou alguns estados iniciais. Uma sub-solução do problema é construída outras previamente encontradas. Soluções de programação dinâmica tem uma complexidade polinomial que assegura um tempo muito menor do que outras técnicas como backtracking, força bruta, etc..

A idéia central da programação dinâmica é evitar trabalhos repetidos ao lembrar resultados parciais e esse conceito pode ser aplicado em muitos dos problemas da vida real. A programação dinâmica permite que problemas complexos sejam resolvidos em tempo O(n²) ou O(n³) para os quais leigos levariam tempo exponencial.

(C) Algoritmo Guloso.

Algoritmo guloso ou greedy (ganancioso), é uma estratégia de solução de problemas em que escolhe-se, a cada iteração, o objeto mais apetitoso, que vê pela frente. O objeto escolhido passa a fazer parte da solução que o algoritmo constrói.

Um algoritmo guloso é míope: ele toma decisões com base nas informações disponíveis nas iterações correntes, sem olhar as consequências que essas decisões terão no futuro. Um algoritmo guloso jamais se arrepende ou volta atrás, as escolhas feitas são definitivas.Embora alguns algoritmos gulosos pareçam obviamente corretos, a prova de sua correção é, em geral, muito sutil. Para compensar, algoritmos gulosos são muito rápidos e eficientes.

Programa Algoritmo Guloso: AALista2AlgoritmoGuloso

(D) Backtracking.

Backtracking é um tipo de algoritmo que representa um refinamento da busca por força bruta, em que múltiplas soluções podem ser eliminadas sem serem explicitamente examinadas. O termo foi cunhado pelo matemático estado-unidense D. H. Lehmer na década de 1950.

Programa Backtracking: AALista2Backtracking

6-Programa para o problema de multiplicação de matrizes: AALista2MultiplicaçãoDeMatrizes

7-

a -

Na teoria da complexidade computaciona[l](https://pt.wikipedia.org/wiki/Complexidade_computacional), o problema de satisfabilidade booleana (SAT) foi o primeiro problema identificado como pertencente à classe de complexidade NP-completo. O problema de satisfatibilidade booleana é o problema de determinar se existe uma determinada valoração para as variáveis de uma determinada fórmula booleana tal que esta valoração satisfaça esta fórmula em questão.

O problema SAT foi o primeiro problema NP-completo conhecido, como demonstrado por Stephen Cook em 1971 (teorema de Cook). Até essa época, o conceito de um problema NP-completo nem existia. O problema SAT continua NP-completo mesmo se todas as fórmulas estiverem na forma normal conjuntiva com 3 variáveis por cláusula (3-FNC), gerando o problema 3SAT.

A complexidade de SAT é um limite inferior para a complexidade do problema P porque se o problema P for resolvido em tempo polinomial, o problema SAT também poderá ser resolvido em tempo polinomial.

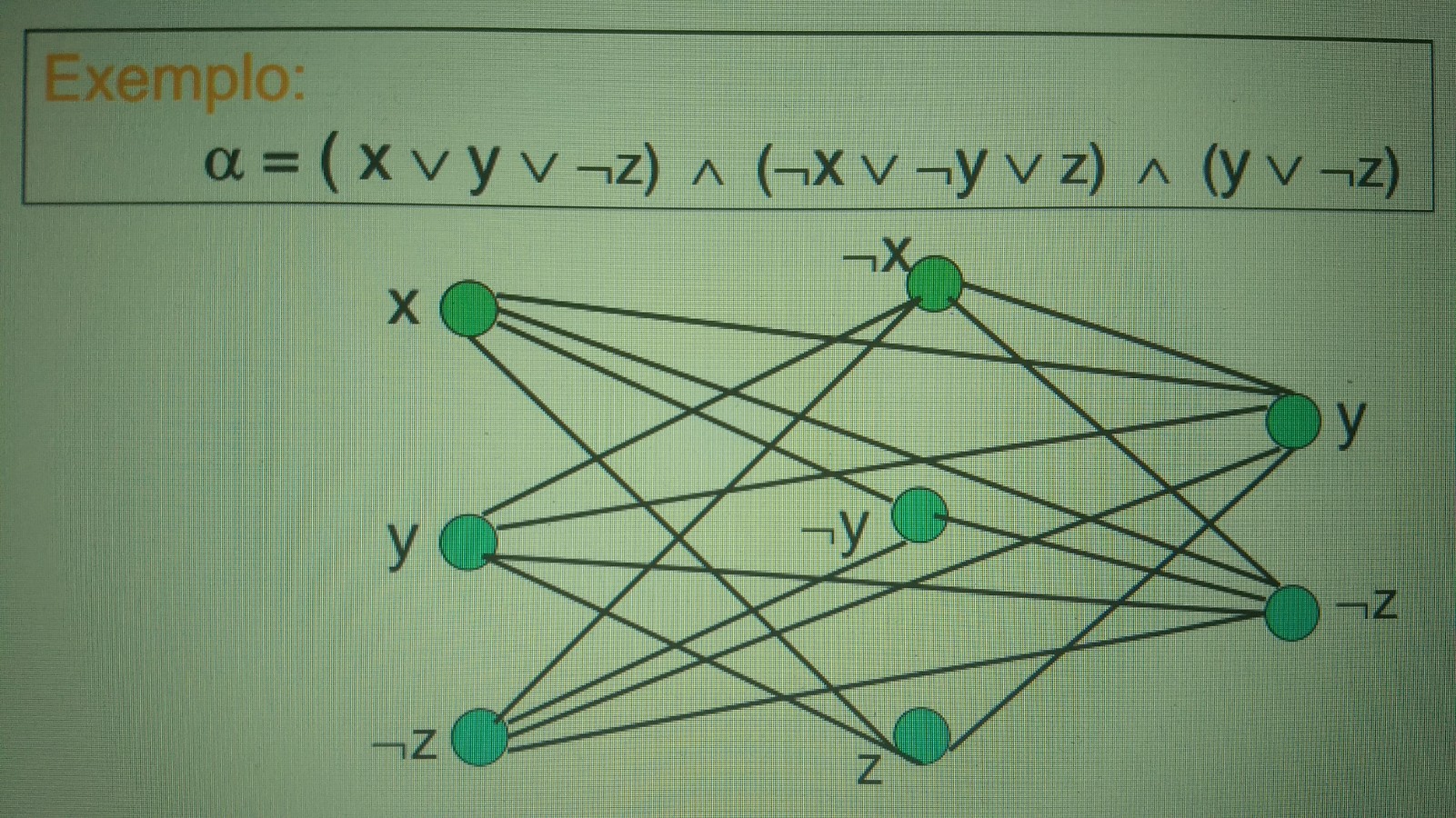
Redução de cook:

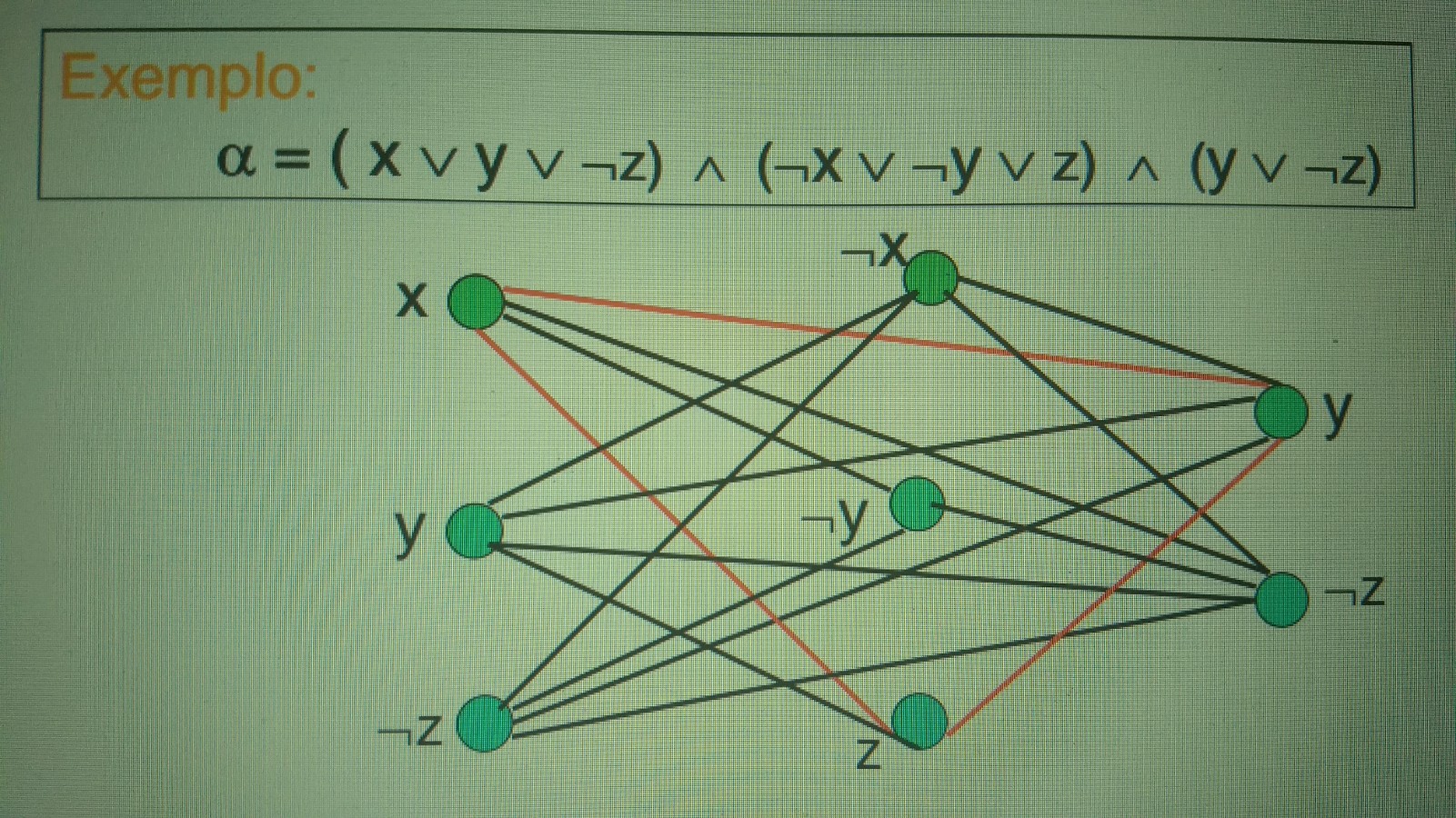
Entrada: Expressão booleana , na Forma Normal Conjuntiva (FNC), ou seja, uma conjunção de disjunções.

Saída: Existe uma valoração das variáveis de de forma que seja verdadeira?

Exemplo:

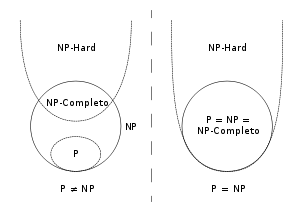
a = ( X v Y v ¬Z) ^ (¬X v ¬Y v Z) ^ (Y v ¬Z)





*Exemplo de redução de Cook a partir de uma sentença lógica*

b -



* classe p: Problemas com um algoritmo eficiente para achar a solução. É uma classe fundamental, que contém todas as decisões de problema que podem ser resolvidos usando uma máquina determinística de Turing usando uma quantidade polinomial de tempo, ou tempo polinomial.

exemplo:sequência de números crescentes.

* classe np: Problemas com um algoritmo eficiente para verificar provas, certificados. Algumas vezes não se sabe a forma eficiente para resolver um problema de decisão, no entanto se alguém nos dá a resposta e a prova, nós temos como verificar se a resposta está correta, somente checando a prova para ver se é uma prova válida. Essa é a idéia por trás da classe np.

exemplo: fatoração, equação do segundo grau.

* classe np-completo: Problemas NP universal. Um problema X é NP-difícil se todos os problemas em NP não são mais difíceis que X. Dito de maneira mais informal: um problema é NP-difícil se for tão difícil quanto qualquer problema em NP. Um problema é NP-completo se for NP-difícil e estiver em NP.

exemplo: soma do subconjunto, mochila valiosa.

* np difícil: Um problema X é NP-difícil se todos os problemas em NP não são mais difíceis que X. Dito de maneira mais informal: um problema é NP-difícil se for tão difícil quanto qualquer problema em NP.

exemplo: caminho curto, subsequência crescente longa.

Questão extra -

Uma série de números que tem uma característica especial de regressão, foi exposta no ano de 1202, no livro denominado Líber Abacci (o livro do ábaco), que nele consta também grande quantidade de assuntos relacionados com a Aritmética e Álgebra da época, postos por Leonardo de Pisa (1175 – 1250), que foi posteriormente identificado como Leonardo Fibonacci (filho de Guiliermo Bonacci) e mais recentemente identificado em suas obras apenas pela palavra Fibonacci. Com este seu trabalho e com os outros, Practica Geometriae (1220), Líber Quadratorum (1225) e Flos (1225), ele cooperou de maneira importante para o desenvolvimento matemático na Europa nos séculos seguintes. Posteriormente estes números em série ficaram conhecidos como Sequência de Fibonacci e deles foram extraídos conclusões até então não imagináveis.

Em todo o Universo está presente “a marca” ou a presença de Deus responsável pelo fenômeno simétrico da natureza. Ela é constatada através da Proporção Áurea proveniente da Sequência de Fibonacci, que se mostra presente como “o sinal divino” em toda a natureza. Portanto, nas flores, árvores, ondas, conchas, furacões, no do rosto simétrico do ser humano, em suas articulações, seus batimentos cardíacos e em seu DNA. Também na refração da luz proporcionada pelos elétrons dos átomos, nas vibrações e em outras mais manifestações da Criação como nas galáxias do Universo imensurável. A relação da Série de Fibonacci e o Número de Ouro em sequência numérica e geométrica parece de modo significativo ser “a marca” de um “Designer” – a “impressão digital” de sua Creação.

O Número de Ouro ou Proporção Áurea é uma constante real algébrica conhecida pela letra grega (PHI) extraída da Sequência de Fibonacci. Ele que possui o valor aproximado de 1,618, que está envolvida em toda a natureza ao buscar o crescimento e que é utilizada nas artes de um modo geral como uma proporção buscando o harmônico, não deve ser confundida com o número Pi π com o valor numérico aproximado de 3,14, que pertence aos números irracionais.

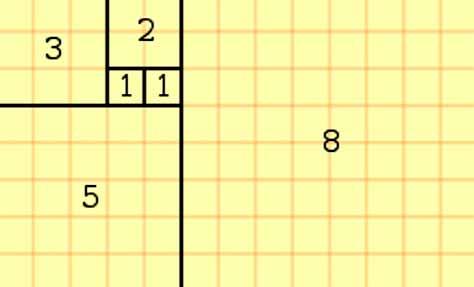
A Proporção Áurea é também chamada de Seção Áurea, Razão Áurea, Razão de Ouro, Divina Proporção, Proporção em Extrema Razão e Divisão de Extrema Razão.

Justamente por estar envolvido em toda a natureza relacionando-se à dinâmica do crescimento, que faz o Número de Ouro tão frequente e, justamente por causa desta frequência como “a marca” de Deus que ele ganhou o status de muito especial, sendo alvo de pesquisadores, de cientistas e também, de artistas e de escritores.

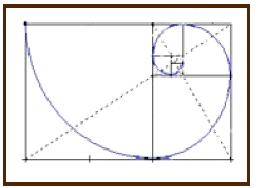
O Número de ouro que é devido aos contextos em que está inserido, está envolvido em crescimentos biológicos e ainda observado nas pinturas, nas partituras e na arquitetura, etc. Como “o selo” de Deus, ele está sempre presente “no construir” a Harmonia Universal. E o fato de ser encontrado através de desenvolvimento matemático, o torna neste sentido ainda mais fascinante.

Esta “marca” vista na proporção áurea é proveniente da Série ou Freqüência de Fibonacci. Nesta sucessão matemática cada número é obtido somando os dois últimos dígitos, ou seja, 1, 1 (1+1) 2, (2+1) 3, (3+2) 5, (5+3) 8, (8+5) 13, (13+8) 21 … continuando em uma sequência infinita.

Utilizando-se deste sistema numérico para construir um retângulo com dois números interligados desta sequência, forma-se o chamado Retângulo de Ouro, que é considerado o formato retangular mais belo e apropriado de todos. E o Retângulo de Ouro quando é divido por quadrados proporcionais à Sequência de Fibonacci, ele alarga o seu conjunto consoante a sucessão de Fibonacci.



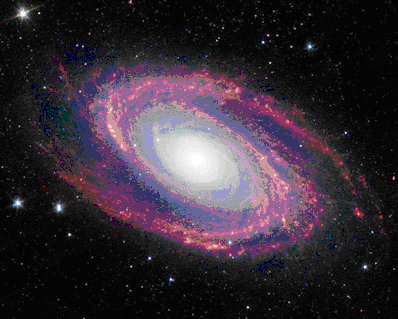
*Retângulo de Fibonacci*



*Espiral de Fibonacci*



*Flor com simetria parecida com a proporção áurea*



*Galáxia em espiral com simetria parecida com a proporção áurea*



*Parte da estrutura óssea humana com simetria similar à proporção áurea*