/\*

2ª Lista

Disciplina: Análise de Algoritmo – DCC606

Aluno: Arthur de Melo Gerônimo

Matrícula: 1201124431

Questão 05

\*/

**[QUESTÃO – 05]**

**Defina, explicando as principais características e exemplifique:**

**(A) Enumeração explícita x implícita.**

**Resposta:**

**Enumeração:**

Enumeração é quando temos elementos de um conjunto finito com valores atribuídos, normalmente esses números são atribuídos como inteiros.

**Explicita x Implícita:**

Um exemplo de enumeração explicita x implícita. Tendo um baralho espalhado na mesa, o jogador já tem conhecimento do valor de cada carta, não irá precisar virar carta alguma para saber seu

**(B) Programação Dinâmica.**

**Resposta:**

Utiliza um esquema de enumeração de soluções que visa, através de uma abordagem de divisão e conquista

(decomposição), minimiza o montante de computação a ser feito. A sua aplicação é indicada quando há casos em que um mesmo sub-problema aparece diversas vezes ao longo do processo, onde a decomposição pura e simples é incapaz de reconhecer este fato, pois os sub-problemas são resolvidos uma vez só e reutiliza a solução toda vez que o mesmo aparecer novamente. Programação dinâmica é programação recursiva usando tabelas, ao invés de resolver recursivamente, ela resolve de forma sequencial armazenando as soluções em uma tabela, assim toda vez que uma solução para um subproblema for necessária ela já vai estar na tabela. Programação dinâmica é muito utilizada em problemas de divisão e conquista que tendem a ter um número de subproblemas exponenciais, mas esses sub-problemas se repetem frequentemente, assim a tabela irá reduzir bastante o tempo de execução.

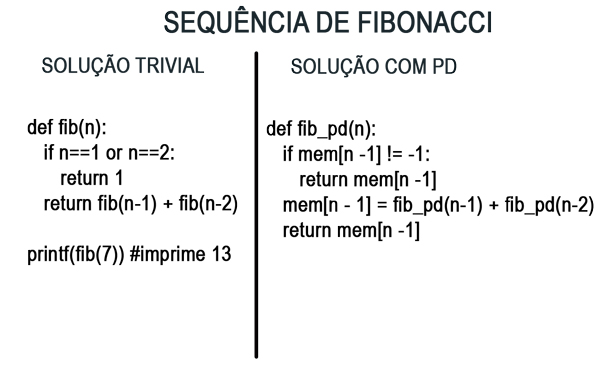


Figura 1 - Fibonacci com Programação Dinâmica

**(C) Algoritmo Guloso.**

**Resposta:**

Para resolver um determinado problema o algoritmo Guloso sempre escolhe a alternativa que mais lhe parece promissora naquele instante. Ele nunca reconsidera essa decisão, portanto, uma escolha que foi feita nunca é revista, não há o chamado backtracking. Essa alternativa mais promissora é estabelecida a priori.

Embora algoritmos gulosos pareçam óbviamente corretos, a prova de sua correção é, em geral, muito sutil. Para compensar, algoritmos gulosos são muito rápidos e eficientes.

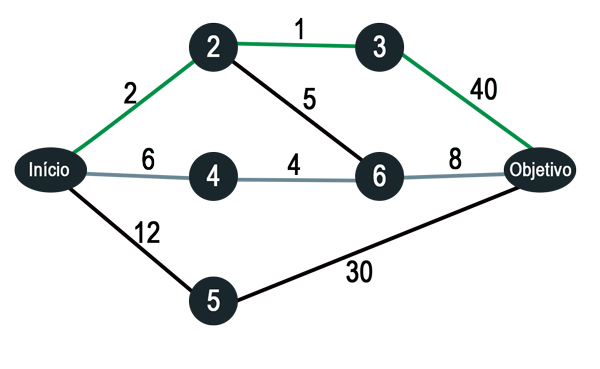


Figura 2 - Caminho percorrido pelo algoritmo guloso

Perceba no exmplo que o caminho com as linhas de cor azul teriam o menor custo, logo, seria o melhor caminho, porém, o algoritmo guloso foi escolhendo o caminho com o menor valor, logo, gerou um caminho com um custo elevado.

**(D) Backtracking.**

**Resposta:**

Backtracking é um modo metódico de tentar várias sequências de decisões, até encontrar uma que funcione. Conhecido por ser um algoritmo de força bruta, ele toma várias decisões entre várias possibilidades, onde não há informações suficiente para saber qual decisão tomar, mas dentre umas dessas decisões, uma é a satisfatória. O algoritmo usa recursão para dividir o processo em várias subtarefas, assim pode explorá-las e gerar uma árvore de tentativas.

A ideia para o algoritmo é decompor o processo em um número finito de sub-tarefas parciais, que são expressas de forma recursiva. A partir desse ponto, o algoritmo explora as sub-tarefas exaustivamente. A construção de um solução é obtida através de tentativas da árvore de sub-tarefas.

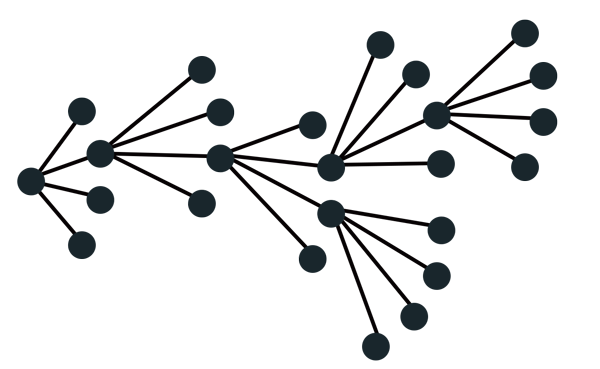


Figura 3 - O processo de tentativa gradualmente constr

**Referências:**

ROMAN, Norton T.. **Algoritmos Gulosos**. 2013. Disponível via URL em: <http://www.each.usp.br/digiampietri/SIN5013/13-algoritmosGulosos.pdf>. Acesso em 12 de julho de 2017.

ROMAN, Norton T.. **Tentativa e Erro (Backtracking).** 2012. Disponível via URL em: < http://www.each.usp.br/digiampietri/SIN5013/12-tentativaEErro.pdf >. Acesso em 12 de julho de 2017.

CASTRO, Marcos. **Programação Dinâmica**. 2016. Disponível via URL em: < https://www.slideshare.net/mcastrosouza/programao-dinmica-57438938>. Acesso em 12 de julho de 2017.