

Inteligência Computacional

Rafael D. Ribeiro, M.Sc.
rafaeldiasribeiro@gmail.com
<http://www.rafaeldiasribeiro.com.br>

Inteligência Computacional

Métodos Revogáveis de Busca

- Busca em profundidade
- Backtracking
- Busca em largura
- Busca em profundidade iterativa
- Busca ordenada

Inteligência Computacional

Busca em profundidade

- A busca em profundidade se caracteriza por priorizar os **nós em níveis mais profundos da árvore**.
- Inicialmente, o método expande o **nó raiz**, isto é, **gera todos os seus filhos**.
 - Se um desses **filhos for a solução** do problema, então o **método é interrompido**.
 - Caso nenhum deles seja a solução, **um deles é escolhido segundo algum critério para ser expandido e seus filhos são testados para saber se algum representa a solução**.

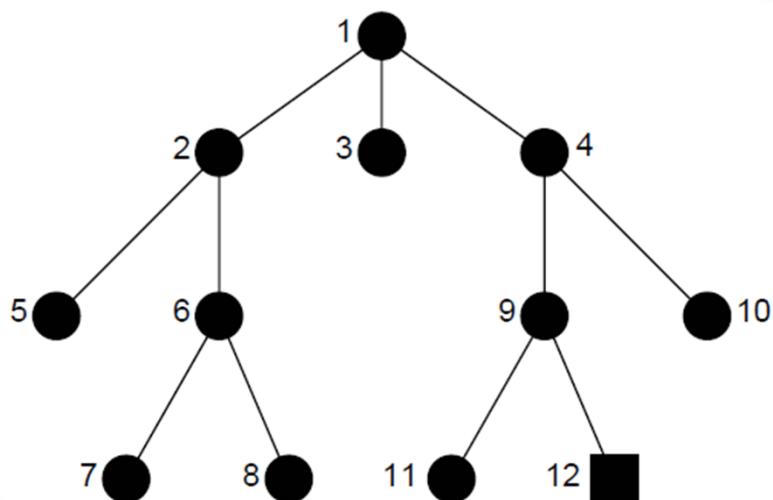
Inteligência Computacional

Busca em profundidade

- Em geral, este critério seleciona o **nó** que está em **maior profundidade** e mais à **esquerda** na árvore de busca.
- O processo continua até que alguma solução seja encontrada ou um nó selecionado para a expansão não possua filhos (nó folha).
- Neste caso, **os irmãos deste nó folha são verificados até que se encontre um em que a busca possa continuar em níveis mais profundos**, isto é, até se obter um nó que possa ser expandido.
- Se **todos os irmãos também forem nós folhas**, o método **retorna ao nível anterior** da árvore e passa a **avaliar os irmãos** do nó expandido deste nível, repetindo-se todo o procedimento.

Inteligência Computacional

Busca em profundidade



Inteligência Computacional

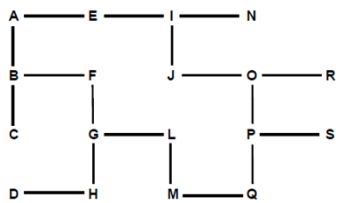
Algoritmo básico de busca em profundidade

```

resposta = nulo
pilha-de-abertos = estado inicial
lista-de-fechados = nulo
sucesso = falso
enquanto (sucesso = falso) e (pilha-de-abertos ≠ vazio) faça
  nó-candidato = elemento do topo da pilha-de-abertos
  remova o elemento do topo da pilha-de-abertos
  coloque-o em lista-de-fechados
  expanda nó-candidato
  se algum filho de nó-candidato é a solução
    então
      sucesso = verdadeiro
      resposta = filho-solução de nó-candidato
  senão
    coloque o(s) nó(s) filho(s) no topo de pilha-de-abertos
fim-se
fim-enquanto
retorna sucesso e resposta
  
```

Inteligência Computacional

Problema do Labirinto



Método Busca em Profundidade

Entrada: A Saída: S

Regras:

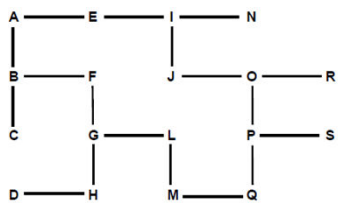
- Regra 1: ↓
- Regra 2: ←
- Regra 3: ↑
- Regra 4: →

Estratégia:
As regras são aplicadas na ordem acima

Inteligência Computacional

Problema do Labirinto

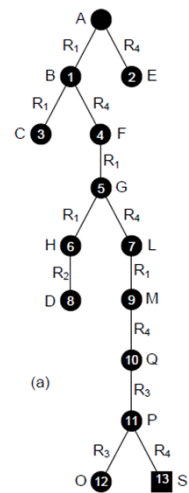
Solução – Método Busca em Profundidade:



Entrada: A Saída: S

Regras:

- Regra 1: ↓
- Regra 2: ←
- Regra 3: ↑
- Regra 4: →



Inteligência Computacional

Problema dos jarros de água

- Existem dois jarros inicialmente vazios. Um possui capacidade igual a 3 litros e o outro igual a 4 litros.
- Ambos podem ser enchidos completamente utilizando uma torneira e podem também ser esvaziados, despejando a água em um ralo.
- Além disso, a água presente em um jarro pode ser passada para o outro.
- Os jarros não possuem marcações e não é permitido o uso de qualquer instrumento de medida. Deseja-se colocar exatamente dois litros de água no jarro maior.
- Pode-se representar as quantidades de água presentes nos dois jarros pelo par ordenado (x,y) , em que x é a quantidade de água no jarro menor e y a quantidade no jarro maior.
- O estado inicial (ambos vazios) é o par $(0,0)$ e o objetivo é encontrar um par do tipo $(x,2)$, isto é, dois litros no jarro maior e qualquer quantidade no jarro menor.

Inteligência Computacional

Problema dos jarros de água

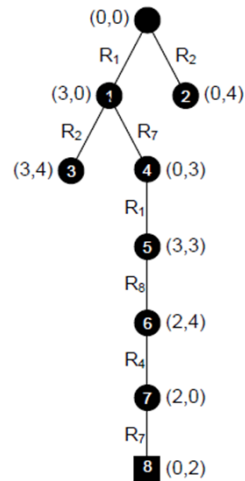
- As regras que definem as ações permitidas são exibidas a seguir, observando que a estratégia de aplicação das mesmas é a ordem de apresentação:.

- R₁: Se $x < 3$, então $(3,y)$. // regra para encher o jarro menor
 R₂: Se $y < 4$, então $(x,4)$. // regra para encher o jarro maior
 R₃: Se $x > 0$, então $(0,y)$. // regra para esvaziar o jarro menor
 R₄: Se $y > 0$, então $(x,0)$. // regra para esvaziar o jarro maior
 R₅: Se $y > 0$ e $x+y \leq 3$, então $(x+y,0)$. // regra para passar toda a água do jarro maior para o menor
 R₆: Se $y > 0$ e $x+y > 3$, então $(3,y - (3-x))$. // regra para passar parte do maior para o menor
 R₇: Se $x > 0$ e $x+y \leq 4$, então $(0,x+y)$. // regra para passar toda a água do jarro menor para o maior
 R₈: Se $x > 0$ e $x+y > 4$, então $(x - (4-y),4)$. // regra para passar parte do menor para o maior

Inteligência Computacional

Problema dos jarros de água

Solução – Método Busca em Profundidade:



Inteligência Computacional

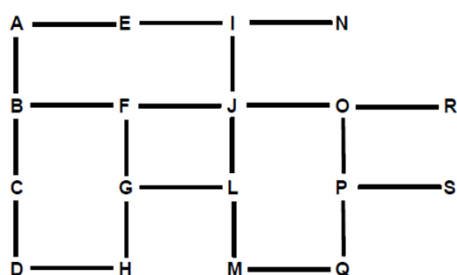
- Existem casos em que a **árvore** de busca possui **profundidade infinita** e, então, a **busca em profundidade** pode continuar a testar os nós mais profundos de um ramo que não contenha alguma solução e ser incapaz de retornar e testar outros ramos. Assim, **ela não é uma busca completa**.
- Por isso, muitas implementações deste método incorporam o conceito de **limite de profundidade**, sendo chamadas de **busca em profundidade limitada**. Deste modo, a árvore não é explorada além deste limite pré-estabelecido e a busca retorna a níveis anteriores quando um **nó é identificado como folha ou quando o limite de profundidade foi alcançado**. É necessário ter muito cuidado na determinação deste limite para **evitar que somente haja soluções em profundidades maiores que ele**.

Inteligência Computacional

- Em árvores finitas, a busca em profundidade é sempre capaz de encontrar alguma solução, se esta existir, mas não garante que ela seja a melhor. Portanto, não é uma técnica ótima.

Inteligência Computacional

Utilizando o Método de Busca em Profundidade, resolva:



Regras

Regra 1: →

Regra 2: ↓

Regra 3: ↑

Regra 4: ←

Partida: N

Destino: Q

Estratégia:

As regras são aplicadas na ordem acima

Inteligência Computacional

Utilizando o Método de Busca em Profundidade, solução:

```

graph TD
    0((0 N)) -- R2 --> 1((1 I))
    1 -- R4 --> 3((3 E))
    1 -- R2 --> 2((2 J))
    2 -- R4 --> 6((6 F))
    2 -- R1 --> 4((4 O))
    2 -- R2 --> 5((5 L))
    4 -- R1 --> 7((7 R))
    4 -- R2 --> 8((8 P))
    8 -- R1 --> 9((9 S))
    8 -- R2 --> 10((10 Q))
    
```

Regras

Regra 1: →

Regra 2: ↓

Regra 3: ↑

Regra 4: ←

Partida: N
Destino: Q

Inteligência Computacional

Utilizando o Método de Busca em Profundidade, resolva:

Regras

Regra 1: →

Regra 2: ↓

Regra 3: ↑

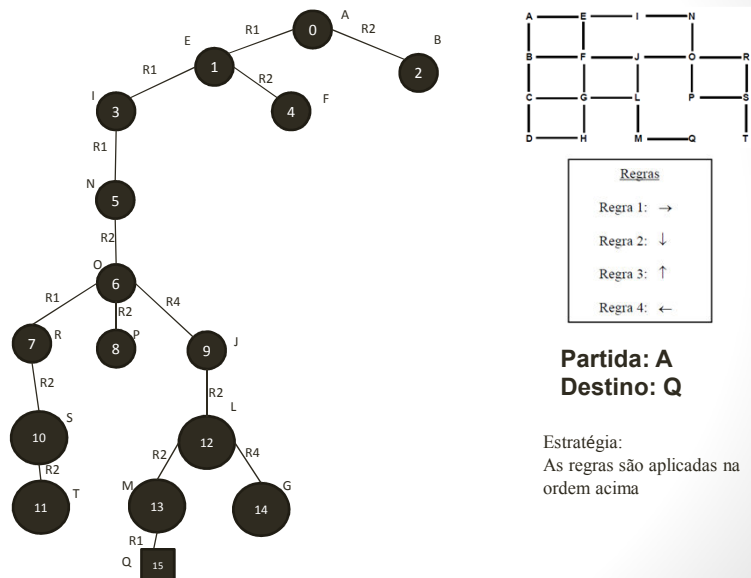
Regra 4: ←

Partida: A
Destino: Q

Estratégia:
As regras são aplicadas na ordem acima

Inteligência Computacional

Utilizando o Método de Busca em Profundidade, solução:



Inteligência Computacional

- Bibliografia utilizada para estas notas de aula:
 - Russel, Stuart J. Inteligência Artificial. Stuart J. Russel. Rio de Janeiro. Elsevier. 2004
 - Notas de Aula do Prof. ROGÉRIO ESPÍNDOLA, disponível na Biblioteca Virtual de docentes – SIA – Estácio, para a disciplina de Inteligência Computacional 1