

**Departamento de Ciência de Computadores - FCUP**  
**Primeiro Teste de Inteligência Artificial / Sistemas Inteligentes**  
**(Duração: 1 hora)**

Data: 20 de Março de 2017

1) Dada a árvore de procura da Figura 1, mostre a sequencia de nós visitados pelo algoritmo IDA\* nesta árvore. Números nas arestas denotam o custo real do caminho entre pares de nós. O número que aparece dentro de um nó corresponde à heurística associada a este nó.

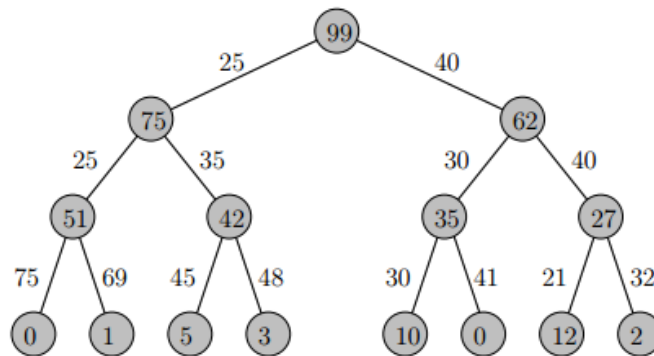


Figura 1: Árvore de procura para a pergunta 1

**Sequencia de visita dos nós para cada f-limit**

```
fmin = 99
  99(99) - 75(100) - 62(102)
fmin = 100
  99(99) - 75(100) - 51(101) - 42(102) - 62(102)
fmin = 101
  99(99) - 75(100) - 51(101) - 0(125) - 1(120) - 42(102) - 62(102)
fmin = 102
  99(99) - 75(100) - 51(101) - 0(125) - 1(120) - 42(102) - 5(110) - 3(111) -
  62(102) - 35(105) - 27(107)
fmin = 105
  99(99) - 75(100) - 51(101) - 0(125) - 1(120) - 42(102) - 5(110) - 3(111) -
  62(102) - 35(105) - 10(110) - 0(111) - 27(107)
fmin = 107
  99(99) - 75(100) - 51(101) - 0(125) - 1(120) - 42(102) - 5(110) - 3(111) -
  62(102) - 35(105) - 10(110) - 0(111) - 27(107) - 12(113) - 2(114)
fmin = 110
  99(99) - 75(100) - 51(101) - 0(125) - 1(120) - 42(102) - 5(110) - 3(111) -
  62(102) - 35(105) - 10(110) - 0(111) - 27(107) - 12(113) - 2(114)
fmin = 111
  99(99) - 75(100) - 51(101) - 0(125) - 1(120) - 42(102) - 5(110) - 3(111) -
  62(102) - 35(105) - 10(110) - 0(111)
```

Melhor caminho encontrado: 99 - 62 - 35 - 0, com custo mínimo 111

2) Sendo h1 e h2 duas heurísticas admissíveis, indique quais das alternativas abaixo são também admissíveis:

- (A)  $0.1 \cdot h_1 + 0.9 \cdot h_2$ : esta soma é sempre menor do que  $h_1$  ou  $h_2$ , portanto é admissível.
- (B)  $\min(h_1, h_2)$ : se  $h_1$  é admissível e  $h_2$  são admissíveis,  $h_1$  ou  $h_2$  também são admissíveis.
- (C)  $\min(h_1, 2 \cdot h_2)$ : se  $h_1 < 2 \cdot h_2$ , ficamos com  $h_1$ , que é admissível. Se  $2 \cdot h_2 < h_1$ , ficamos com  $2 \cdot h_2$  e como é menor do que  $h_1$ , também é admissível.
- (D)  $\min(h_1, 0.5 \cdot h_2)$ : mesma razão acima.
- (E)  $h_1 - h_2$ : como pode dar resultado negativo passa a não ser uma heurística (conceito de distância), portanto pode não ser admissível.

3) Execute o algoritmo minimax na árvore da Figura 2. Escreva o valor de cada um dos nós especificados na caixa de texto abaixo.

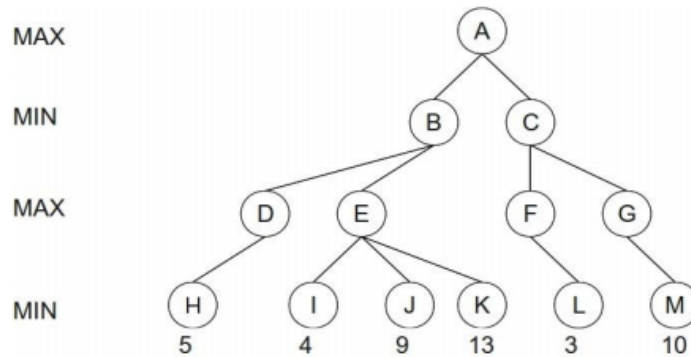


Figura 2: Árvore de jogo para a pergunta 3

A= 5 B= 5 C= 3 D= 5 E= 13 F= 3 G= 10

4) Usando a mesma árvore de jogo da pergunta 3, qual seria a ordem ótima de valores de utilidade e nós para que o algoritmo alpha-beta fizesse a poda máxima? Anote esta ordem nos nós da árvore da Figura 3. Qualquer ordenação em que os nós que são escolhidos pelo max estão em ordem decrescente e os nós que são escolhidos pelo min estão em ordem crescente.

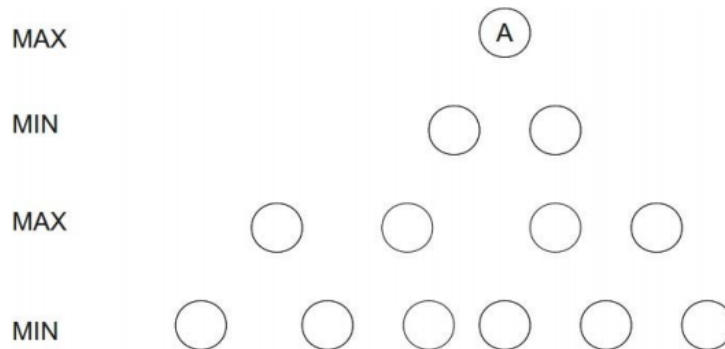


Figura 3: Árvore de jogo para a pergunta 4

5) Escreva um pseudo-código para o algoritmo “simulated annealing”.  
Está no livro texto, 3rd edição, cap. 4, seção 4.1.2

6) Indique como modificaria este algoritmo para que se comportasse como um algoritmo “hill climbing”.  
Basta modificar as seguintes linhas do algoritmo de simulated annealing:

```
if DeltaE > 0 current <- next  
else return current
```

This means that the probability of accepting worse states is zero!

**As duas próximas perguntas correspondem à parte prática do teste. Esta parte deve ser respondida apenas por aqueles que não entregaram os dois primeiros trabalhos.**

### (PRÁTICA 1)

Dado o grafo da Figura 4, explique porque a heurística do nó E não é apropriada para ser utilizada pelo algoritmo A\*.

Por que a distância real de E para G é 14 e a heurística de E diz que deveria ser 15. Este valor de heurística sobrestima o custo real do caminho de E a G, portanto não é admissível.

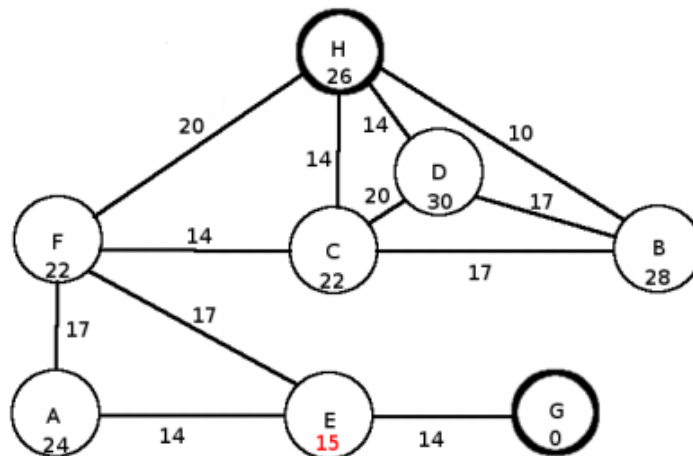


Figura 4: Figura para a pergunta prática 1

### (PRÁTICA 2)

Sejam os dois algoritmos mostrados nas Figuras 5 e 6. O primeiro algoritmo corresponde ao algoritmo geral de busca dado em aula. O segundo, Figura 6, faz uma pequena alteração ao algoritmo geral, onde o teste pela solução (GOAL\_TEST) é apenas efetuado durante a geração dos nós descendentes. Responda qual seria o efeito, em termos de custo computacional, completude e otimalidade desta modificação, para as estratégias:

- (A) Busca em profundidade: o segundo algoritmo, dependendo da ordem em que os descendentes são colocados na lista de nós abertos, poderia encontrar a solução mais cedo, assim que gerar um deles.
- (B) Busca em largura: o algoritmo 2 permitiria gerar menos descendentes de um determinado nível se um deles já fosse solução.
- (C) Busca iterativa em profundidade: o algoritmo 2 pode perder a solução ótima.

**(D)** Guloso (que usa a heurística como função de avaliação): [mesmo que o DFS](#).

**(E)** A\*: [o algoritmo 2 pode perder a solução ótima](#).

```
function GENERAL_SEARCH(problem, QUEUEING_FN) retorna solucao ou falha
  nodes := MAKE_QUEUE(MAKE_NODE(INITIAL_STATE(problem)))
  loop
    if EMPTY?(nodes) then return falha
    node := REMOVE_FRONT(nodes)
    if STATE(node) = GOAL_TEST[problem] then return node
    successors = empty list
    foreach s = successor(node, Operators)
      add(s, successors)
    end
    nodes := QUEUEING_FN(nodes, successors)
  end
```

Figura 5: Figura para a pergunta (PRÁTICA 2)

```
function GENERAL_SEARCH(problem, QUEUEING_FN) retorna solucao ou falha
  nodes := MAKE_QUEUE(MAKE_NODE(INITIAL_STATE(problem)))
  loop
    if EMPTY?(nodes) then return falha
    node := REMOVE_FRONT(nodes)
    successors = empty list
    foreach s = successor(node, Operators)
      if STATE(s) = GOAL_TEST[problem] then return s
      add(s, successors)
    end
    nodes := QUEUEING_FN(nodes, successors)
  end
```

Figura 6: Figura para a pergunta (PRÁTICA 2)

### **(PRÁTICA 3)**

Usando a sua linguagem de programação favorita, escreva o código que verifica se o jogo dos 15 tem solução. Assuma que podemos dar qualquer configuração inicial e qualquer configuração final.

[Devem ter implementado no trabalho 1. Um resumo está nos meus slides sob o título Trab1 \(início da página da disciplina\).](#)