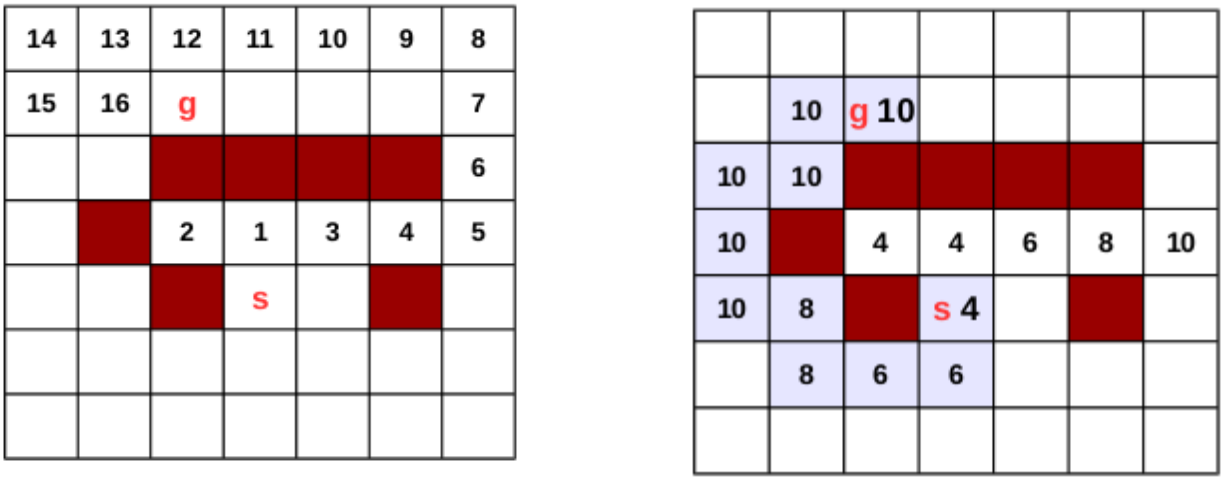
Departamento de Ciência de Computadores - FCUP
Primeiro Teste de Sistemas Inteligentes - Soluções
(Duração: 1 hora e 30 minutos)

Nome: _____

Data: 7 de Abril de 2014 (Soluções publicadas em 27 de Abril)

1) Considere o problema de encontrar um caminho de **s** para **g** na grade da Figura 1. Um robô pode mover-se sobre esta grade horizontalmente ou verticalmente, um quadrado de cada vez (cada passo tem um custo de uma unidade). Nenhum passo pode ser dado nas áreas sombreadas (posições proibidas). Responda ao seguinte:



(a) Busca em profundidade

(b) Busca A*

Figure 1: Soluções para as questões 1(a) e 1(b).

Na busca A*, apenas indiquei alguns dos nós com seus respectivos valores de f. O caminho vai de s a g, em azul. Na busca em profundidade, estão numerados os nós visitados durante a procura pelo caminho de s a g. Aceitei soluções de quem fez a busca de g para s, além de soluções para o A* que seguem pelo lado direito da grade.

(a) Na grade 1(a), numere os nós na ordem em que eles são removidos da lista de abertos, usando uma busca em profundidade, a partir do nó **s** até o nó **g**. A ordem dos operadores é: para cima, para a esquerda, para a direita e para baixo. Assuma que há verificação de ciclos.

(b) Na grade 1(b), numere os nós na ordem em que são retirados da lista de abertos, utilizando o algoritmo A*. A distância de Manhattan deve ser usada como a função heurística. A distância de Manhattan entre dois pontos é a distância na direção **x** mais a distância na direção **y**. Por exemplo, a distância de Manhattan entre **g** e **s** é 4. Qual é o caminho encontrado pela busca A*?

(c) Suponha que o gráfico é estendido infinitamente em todas as direções. Isto é, não há limites na grade, mas **s**, **g** e a área proibida estão nas mesmas posições relativas entre si. Quais os métodos dentre os estudados em aula: busca em profundidade, busca em largura, busca de

custo uniforme, busca profundidade iterativa, busca gulosa e A* já não conseguem encontrar um caminho? Qual seria o melhor método, e por que?

A busca em profundidade nunca vai encontrar uma solução nesta condições, porque assim que chegar à célula numerada com 8, vai seguir sempre acima, e nunca mais vai chegar a g. Qualquer um dos outros métodos estudados encontraria a solução ótima, incluindo a busca gulosa com verificação de ciclos.

2) Descreva e resolva o problema de criptoaritmética SEND + MORE = MONEY, como satisfação de restrições. Cada letra representa um dígito diferente, o dígito mais à esquerda de um número não pode ser zero e a soma deve estar correta de acordo com a equação. A base é 10. Neste exemplo, sabemos que Y=(D+E) mod 10, que E=(N+R+(D+E) / 10)) mod 10 e assim por diante.
Descrição:

- Conjunto de variáveis: V = S,E,N,D,M,O,R,Y
- Domínio: S=1..9, M=1..9, M=1..9, E=N=D=O=R=Y=0..9
- Restrições:
  - $Y = (D + E) \bmod 10$
  - $E = (N + R + (D + E)/10) \bmod 10$
  - $N = (E + O + (N + R + (D + E)/10)/10) \bmod 10$
  - $O = (S + M + (E + O + (N + R + (D + E)/10)/10)/10) \bmod 10$
  - $S \neq E \neq N \neq D \neq M \neq O \neq R \neq Y$
- Encaminhamento da solução:
  - escolher uma variável
  - escolher um valor do domínio desta variável
  - podar valores das outras variáveis afetadas
  - continuar...
  - Por exemplo, $D = 1, Y = (1 + E) \bmod 10$, além disto, remover todos os valores 1 dos domínios das outras variáveis

3) No jogo de soma zero mostrado na Figura 2, qual é o resultado esperado quando se utiliza os algoritmos (assuma que o nó raiz é de MAX):

(a) minimax: 7

(b) corte alfa-beta (além do resultado esperado, indique quais dos nós não precisariam ser visitados): 7

Nós não visitados (13): s,k,v,w,y,a1,g,n,o,a2,a3,a4,a5

4) O jogo do galo, como já conhecem, joga-se em um tabuleiro 3x3, onde dois jogadores, X e O, marcam, alternadamente, as posições ainda não ocupadas. X ganha se, em qualquer momento

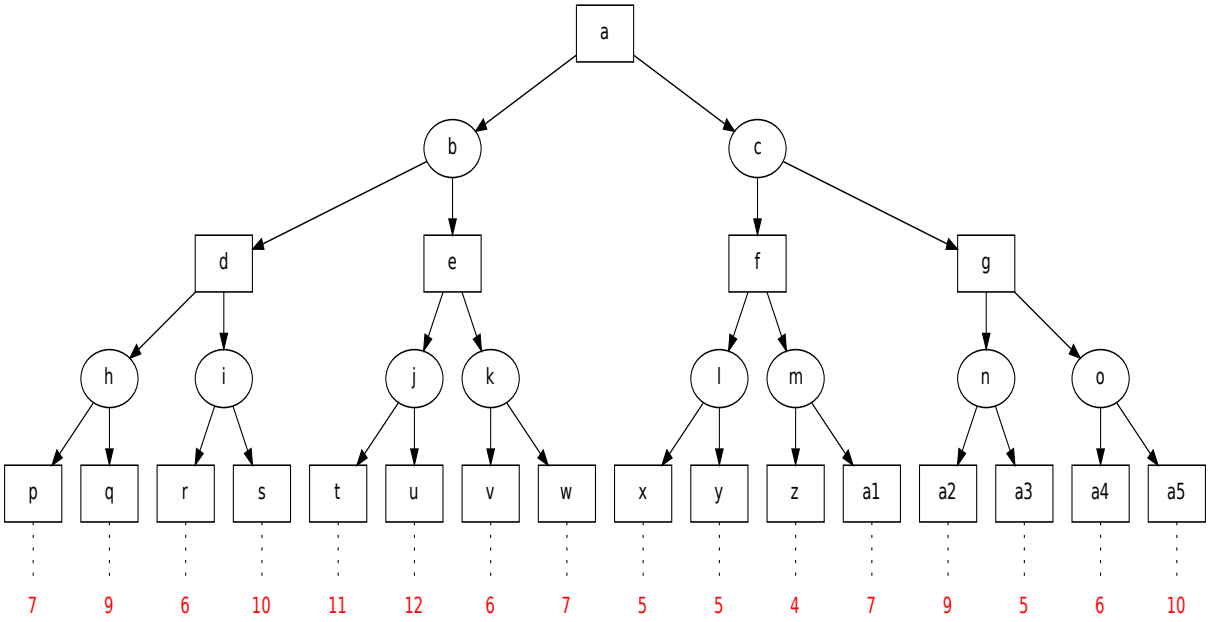


Figure 2: Árvore de jogo para a questão 3

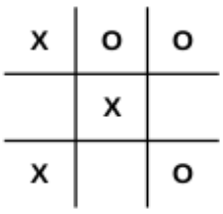


Figure 3: Exemplo de estado do jogo do galo para a questão 4

do jogo, completar uma linha, coluna ou diagonal do tabuleiro. Por exemplo, a Figura 3 mostra um exemplo, onde X pode ganhar se jogar na primeira coluna (da esquerda para a direita) e na segunda linha.
Suponha que duas pessoas, João e Maria precisam escrever um programa para decidir se uma determinada configuração do tabuleiro é vencedora para o X.
João decide fazer a representação do tabuleiro como uma lista de 3 linhas. Por exemplo, usa a representação abaixo para o tabuleiro da Figura 3, onde B representa um espaço não ocupado:

$$[[X, O, O], [B, X, B], [X, B, O]]$$

Maria percebe que as posições deste jogo podem ser representadas de acordo com o quadrado mágico, mostrado na Figura 4.

6	7	2
1	5	9
8	3	4

Figure 4: Quadrado mágico para a questão 4

O jogo então se resume a cada jogador escolher um dígito. E termina quando qualquer um dos dois jogadores tiver uma soma de três números consecutivos igual a 15.
Maria representa então o seu tabuleiro na forma de lista abaixo, onde cada posição da lista corresponde a um número do quadrado mágico. Por exemplo, o número 9 no quadrado mágico aparece como B na Figura 3. Portanto, a entrada 9 da lista abaixo corresponde ao símbolo B.

$$[B, O, B, O, X, X, O, X, B]$$

(a) Para cada uma das duas representações, escreva um pseudo-código que determine se X venceu.

Há várias soluções possíveis. Para quem respondeu que a sua representação era similar a uma das apresentadas (João ou Maria), eu estava esperando que fizesse um pseudo-código similar ao código implementado no seu próprio trabalho.

```
João: (Python-like, very straightforward solution)

# variable board is a list of lists
# check rows
for each row in board if row = ["X","X","X"] return "X wins"

# check columns
for i=0; i<3; i++ {
   count=0
   for j=0; i<3; j++ {
     if board[j][i] = "X" count++
   if count = 3 return "X wins"
```

```
}

# check first diagonal
count=0
for i=0; i<3; i++
   if board[i][i] = "X" count++
if count = 3 return "X wins"

# check second diagonal
count=0
for i=0; i<3; i++
   if board[i][2-i] = "X" count++
if count = 3 return "X wins"

return "X does not win"
```

Maria: (Python-like, very straightforward solution)

```
# variable board is a list with board symbols X and O
# listX is a list of only "X"
# count is a counter for the magic square values
# c is a counter of occurrences of X
c=count=0
for i=1; i<10; i++ {
   if board[i] = "X" {
     listX[c++]
     count+=i
   }
}
if c = 3 and count = 15 return "X wins"
if c > 3 and c > 15 {
   for i=0; i < c; i++ {
      for j=i+1; j < c; j++ {
         for k=j+1; k < c; k++ {
            sum = listX[i]+listX[j]+listX[k]
            if sum = 15 return "X wins"
            else sum = sum - listX[k] # se este k nao contribui para a
                                      # soma, elimino e passo para o proximo k
         }
         sum = sum - listX[j]         # se este j nao contribui para a
                                      # soma, elimino e passo para o proximo j
      }
      sum = sum - listX[i]            # se este i nao contribui para a
                                      # soma, elimino e passo para o proximo i
   }
}
return "X does not win"
```

(b) Qual é a representação mais fácil de usar em termos de código? Explique porque.

Depende da impplementação que fizeram na alínea (a). Segundo a minha implementação, o pseudo-código para verificar a representação do João é mais facilmente interpretável.

(c) Qual é a representação mais eficiente para determinar uma vitória?

Mais uma vez, depende do pseudo-código que escreveram na alínea (a). Nesta minha implementação, no pior caso, precisamos fazer 20 comparações para a representação do João, enquanto fazemos mais de 20 na representação da Maria (9 no primeiro "for" + no máximo "5" no segundo for mais externo + os testes dos "for" mais internos). Portanto, o código associado à representação do João, além de ser mais facilmente interpretável, também pode ser o mais eficiente (no pior caso).

(d) A representação que usou na sua implementação é similar a alguma das duas representações apresentadas? Descreva a sua representação.

Depende do que implementaram no trabalho e do que responderam em relação à similaridade com a representação do João ou da Maria