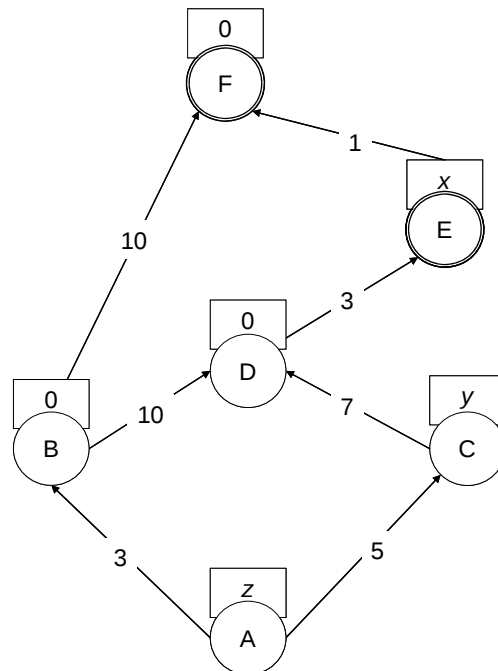


GRUPO I

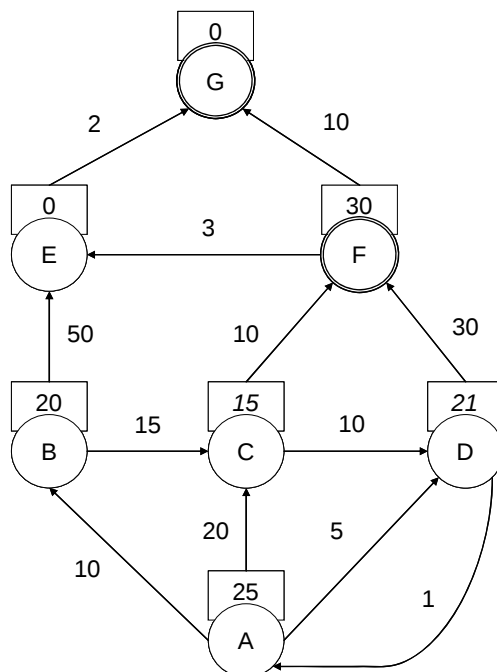
I.1) Considere o seguinte grafo de estados de um problema de procura. Os valores apresentados nos arcos correspondem ao custo do operador (acção) respectivo, enquanto os valores nos rectângulos correspondem ao valor da heurística. Os estados para os quais a função de teste objectivo retorna verdadeiro são o E e o F. Não se representa os nomes dos operadores, correspondendo cada arco a um operador distinto. Indique o intervalo de valores possíveis para x , y e z de modo a que a função heurística seja admissível.



I.2) Suponha que num problema de procura o valor da heurística é sempre nula e que os custos dos operadores são positivos. Justifique que neste caso a procura em árvores A* coincide com a estratégia de procura cega de custo uniforme. Há garantia de obtenção da solução de menor custo (óptima)?

Espaço de rascunho:

I.3) Considere o seguinte grafo de estados de um problema de procura. Os valores apresentados nos arcos correspondem ao custo do operador (acção) respectivo, enquanto os valores nos rectângulos correspondem ao valor da heurística. Os estados para os quais a função de teste objectivo retorna verdadeiro são o F e o G. Partindo do estado inicial A indique qual a primeira solução obtida pelo algoritmo de procura sôfrega e custo respectivo.

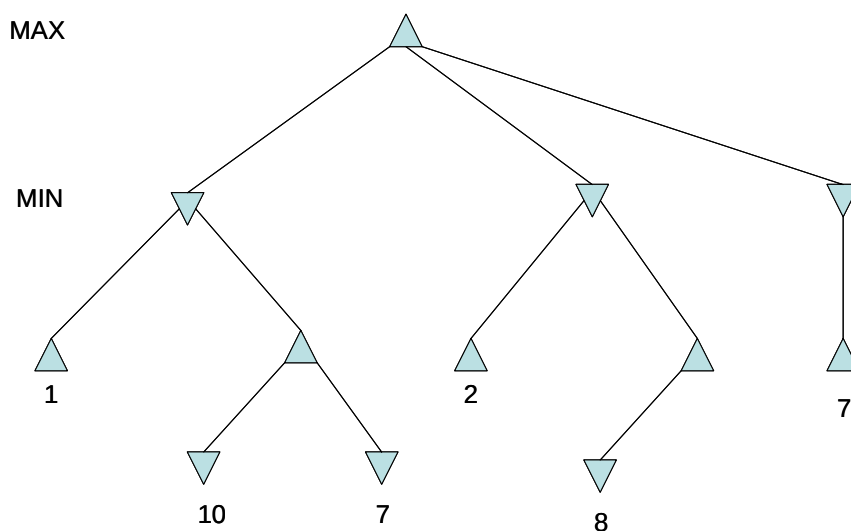


I.4) Sejam A, B e C três variáveis inteiras com domínios, $\{1,2,4\}$, $\{1,2\}$, $\{1,3\}$, respectivamente, sujeitas às restrições binárias indicadas abaixo. Indique qual a variável que seria escolhida pelo algoritmo de procura com retrocesso, admitindo o método da variável mais constrangida com desempate baseado no número de restrições. Justifique a sua resposta.

$$A + B \geq 6.$$

$$B + C \geq 0.$$

I.5) Na figura abaixo poderá encontrar uma árvore de jogo construída pelo algoritmo MINIMAX, em que os nós terminais estão anotados com o valor da função de avaliação. Copie a árvore de jogo para a sua folha de resposta e determine o valor MINIMAX de cada nó da árvore de jogo. Assinale também qual a jogada que o jogador MAX deverá efectuar contra um adversário perfeito.

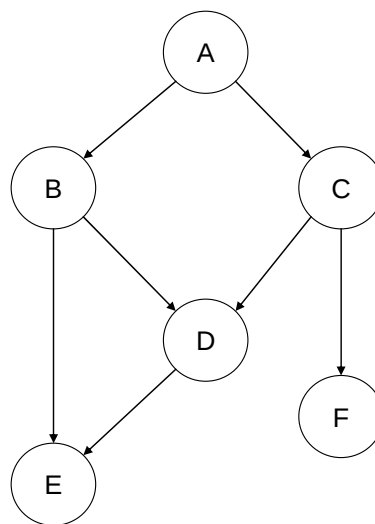


I.6) Recorrendo ao método de resolução em lógica proposicional, demonstre que $(b \wedge \sim c) \vee (a \wedge \sim b)$ é consequência lógica da fórmula $(a \wedge \sim b) \Leftrightarrow (b \Rightarrow c)$. Justifique a sua resposta.

I.7) Considere as variáveis aleatórias booleanas A e B, e a variável aleatória discreta C com domínio $\{1,2,3\}$. A distribuição de probabilidade conjunta $P(A,B,C)$ encontra-se representada na tabela seguinte. Determine o valor de $P(\sim a \vee b \mid C=1)$, sem esquecer de indicar previamente qual o valor de x.

	A=true		A=false	
	B=true	B=false	B=true	B=false
C=1	0,10	0,20	0,05	x
C=2	0,20	0,05	0,05	0,15
C=3	0,01	0,01	0,03	0,15

I.8) Considere a rede Bayesiana constituída apenas por variáveis aleatórias booleanas. Apresente uma expressão simplificada para o cálculo de $P(A \mid D=\text{false})$, na forma de somatório.



I.9) Apresente o Markov blanket para a variável aleatória B da rede Bayesiana da questão **I.8**.

Espaço de rascunho:

I.10) Num problema de aprendizagem de árvores de decisão foram obtidos os seguintes exemplos.

Exemplo	A	B	Classificação
1	1	T	0
2	1	T	0
3	1	F	1
4	2	F	0
5	2	F	0
6	2	T	1
7	3	T	1
8	3	F	1
9	3	F	0

Indique, justificando, qual dos atributos seria escolhido pelo algoritmo DTL como raiz da árvore de decisão. Para facilitar os cálculos, listam-se de seguida alguns valores do logaritmo de base 2 que poderá utilizar, caso entenda necessário.

$\log_2(1/5) = -2,32193$	$\log_2(1/4) = -2,0$	$\log_2(1/3) = -1,58496$	$\log_2(2/5) = -1,32193$	$\log_2(1/2) = -1$
$\log_2(3/5) = -0,73697$	$\log_2(2/3) = -0,58496$	$\log_2(3/4) = -0,41504$	$\log_2(4/5) = -0,32193$	$\log_2(1) = 0$

Espaço de rascunho:

GRUPO II

Seja T um conjunto de tarefas a realizar num sistema multiprocessador com m processadores iguais. Cada tarefa t pertencente a T tem uma duração $d(t)$, um número positivo. O objectivo consiste em atribuir a cada tarefa um processador onde deverá ser executada, minimizando o tempo de conclusão do conjunto de tarefas de T . Essa atribuição (escalonamento) é representada pela função $f: T \rightarrow [1, \dots, m]$ (para cada tarefa t o valor $f(t)$ indica o processador onde t será executada). O tempo de conclusão do escalonamento é dado pela expressão:

$$\max_{i \in [1, \dots, m]} \sum_{t \in T: f(t)=i} d(t)$$

Ou seja, somam-se por processador as durações das tarefas a executar nele, sendo o tempo de conclusão do escalonamento a maior dessas somas. Por exemplo, considere um sistema com 4 processadores e as 10 tarefas:

Tarefa t :	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
Duração $d(t)$:	4	5	2	4	3	6	4	1	3	2

Um escalonamento possível é dado pela função

Tarefa t :	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
$f(t)$:	1	2	2	4	3	4	1	3	3	2

Com tempo de conclusão 10. Gráficamente, o escalonamento pode ser representado por:

Proc.	1	2	3	4	5	6	7	8	9	10
1	t ₁			t ₇						
2	t ₂				t ₃		t ₁₀			
3	t ₅		t ₈	t ₉						
4	t ₄			t ₆						

O tempo disponível num processador é obtido subtraindo a soma das durações das tarefas executadas nele ao tempo de conclusão do escalonamento. Define-se ainda o tempo total disponível como sendo a soma dos tempos disponíveis em cada processador. Para o exemplo, o tempo total disponível é $6=2+1+3+0$, resultando da soma dos tempos disponíveis nos processadores 1, 2, 3 e 4, respectivamente.

II.1) Apresente uma expressão matemática para a dimensão do espaço de procura, em função do número de processadores e número de tarefas a realizar.

II.2) Formule claramente o problema para ser resolvido recorrendo a algoritmos de procura em espaço de estados, indicando a representação de um estado, o estado inicial, teste de estado objectivo e função que devolve os sucessores de um estado. Preste especial atenção à definição do custo do(s) operador(es).

Nota: Não é necessário apresentar código Java.

II.3) Considere cada uma das seguintes funções heurísticas descritas abaixo. Indique quais delas garantem a obtenção de uma solução óptima pelo algoritmo A* para a classe de problemas dada. Justifique sucintamente a sua resposta, quer para os casos de garantia quer para as situações de não garantia.

- A soma das durações das tarefas que faltam escalonar.
- O tempo da tarefa com a menor duração que falta escalonar.
- O tempo da tarefa com maior duração que falta escalonar, subtraída do maior tempo total disponível de processador. Se for negativo, então utiliza-se o valor 0.
- A soma das durações das tarefas que faltam escalonar subtraída do tempo total disponível, tudo dividido pelo número total de processadores. Se o valor for negativo, então a heurística terá o valor 0.
- O máximo de c) e d).

II.4) Proponha uma representação para resolução deste problema através de um algoritmo genético. Sugira operadores de recombinação e mutação passíveis de utilização.

GRUPO III

O puzzle “Pinta Grelha” é jogado num quadrado n por n . Para cada linha e cada coluna é dado o número total de casas pintadas. O objectivo do puzzle consiste em descobrir qual a figura pintada na grelha. Considere-se o seguinte puzzle (à esquerda) e uma solução possível (à direita).

	1	4	6	3	3	3
2						
3						
2						
5						
5						
3						

Problema

	1	4	6	3	3	3
2						
3						
2						
5						
5						
3						

Uma solução

Recorrendo à programação por conjuntos de resposta, irá produzir neste grupo um programa para o sistema Smodels que lhe permitirá resolver este tipo de puzzles. Considere o seguinte programa, capturando alguma da informação enunciada:

<code>linha(1..6). coluna(1..6). casa(L,C) :- linha(L), coluna(C). canto(1,1). canto(1,6). canto(6,1). canto(6,6).</code>	<code>totalLinha(1,2). totalLinha(2,3). totalLinha(3,2). totalLinha(4,5). totalLinha(5,5). totalLinha(6,3).</code>	<code>totalColuna(1,1). totalColuna(2,4). totalColuna(3,6). totalColuna(4,3). totalColuna(5,3). totalColuna(6,3).</code>
---	--	--

O predicado **totalLinha(Lin,Total)** especifica o número **Total** de casas pintadas na linha **Lin** (numeradas de cima para baixo). Similarmente, o predicado **totalColuna(Col,Total)** especifica o número **Total** de casas pintadas na coluna **Col** (numeradas da esquerda para a direita).

III.1) Recorrendo à programação por conjuntos de resposta gere todas as alternativas correspondendo a que cada casa possa estar pintada ou não. Use o predicado **pintada(Lin,Col)** para indicar esse facto num conjunto de resposta.

III.2) Para cada linha e para cada coluna, o número de casas pintadas deve coincidir com os totais dados no problema.

- Apresente uma solução que garanta esta regra do puzzle, para a linha 1 e para a coluna 1.
- Apresente uma solução geral, válida para todas as linhas e colunas, recorrendo aos factos de **totalLinha/2** e **totalColuna/2**. Note que os limites usados na notação de conjuntos do Smodels podem ser variáveis.

III.3) Indique como representaria cada uma das restrições seguintes em programação por conjuntos de resposta, sem recorrer à sintaxe de conjuntos do sistema SMOBELS.

- A casa (1,2) e a casa (2,1) estão pintadas.
- Nenhum dos cantos está pintado (recorra ao predicado **canto/2** para implementar esta restrição).

III.4) Considere os predicados **l/4** e **a/4** definidos por intermédio das regras apresentadas abaixo. Indique o seu significado, assim como o objectivo da restrição de integridade incluída no código.

Nota: A função **abs** do Smodels determina o valor absoluto de um número.

<code>l(L1,C1,L2,C2) :- casa(L1,C1), casa(L2,C2), pintada(L1,C1), pintada(L2,C2), a(L1,C1,L2,C2).</code>	<code>l(L1,C1,L3,C3) :- casa(L1,C1), casa(L2,C2), casa(L3,C3), l(L1,C1,L2,C2), l(L2,C2,L3,C3).</code>
<code>a(L,C1,L,C2) :- casa(L,C1), casa(L,C2), eq(abs(C1-C2),1).</code>	<code>a(L1,C,L2,C) :- casa(L1,C), casa(L2,C), eq(abs(L1-L2),1).</code>
<code>:- casa(L1,C1), casa(L2,C2), pintada(L1,C1), pintada(L2,C2), not l(L1,C1,L2,C2), L1 != L2. :- casa(L1,C1), casa(L2,C2), pintada(L1,C1), pintada(L2,C2), not l(L1,C1,L2,C2), C1 != C2.</code>	