

1º Teste (Sem consulta)

I) [3,5val] Considere o grafo de estados de um problema de procura da Figura 1. Os valores apresentados nos arcos correspondem ao custo do operador (ação) respetivo, enquanto os valores nos retângulos correspondem ao valor de uma heurística (estimativa do custo de chegar desse estado ao objetivo). Em todos os algoritmos cegos e em caso de empate nos algoritmos informados, quando um nó é expandido, assuma que os seus sucessores são sempre colocados na fronteira por ordem lexicográfica do nome do nó, de forma que o nó mais próximo do início do alfabeto seja selecionado antes dos seus irmãos.

Pretende-se encontrar um caminho desde o estado S até G. Para cada um dos seguintes algoritmos, indique **a ordem pela qual os nós são expandidos**:

- procura em profundidade primeiro (em árvores);
- procura em largura primeiro (otimizada);
- procura por aprofundamento progressivo;
- procura de custo uniforme (em grafos);
- procura sôfrega (em grafos);
- A* (em árvore).

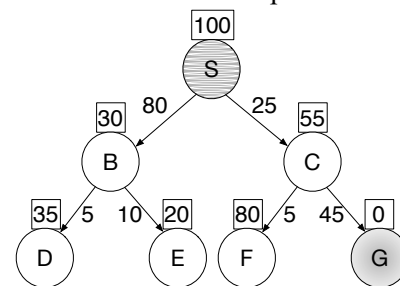


Figura 1

XX

II) [3,5val] Considere o grafo de estados de um problema de procura da Figura 2, onde os valores apresentados nos arcos correspondem ao custo do operador (ação) respetivo. Considere ainda as seguintes três heurísticas, h1, h2 e h3 (em todas elas, o valor nos estados G1, G2 e G3 é igual a 0):

	A	B	C	S
h1	0	0	0	0

	A	B	C	S
h2	6	7	1	7

	A	B	C	S
h3	7	7	1	7

Assuma que **qualquer desempate na escolha de nós é feito aleatoriamente**. Pretende-se encontrar um caminho desde o estado S até G1, G2 ou G3.

Para cada um dos seguintes algoritmos, assinale (com um X) na folha de respostas quais os nós objetivo que **poderiam** ser atingidos:

- procura em profundidade primeiro (em árvores);
- procura em largura primeiro (otimizada);
- procura de custo uniforme (em grafos);
- procura sôfrega (em grafos) com a heurística h1;
- procura sôfrega (em grafos) com a heurística h3;
- A* (em árvore) com h1;
- A* (em árvore) com h2.

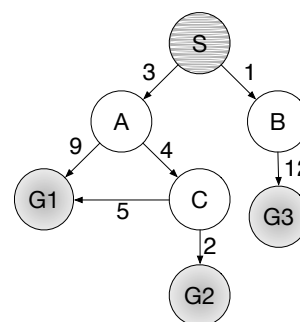


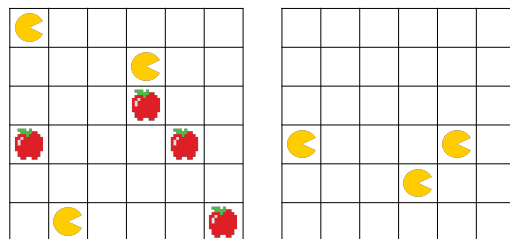
Figura 2

XX

III) [2val] Considere a seguinte variante do algoritmo de procura em profundidade em grafos: registamos o número de vezes que visitámos cada estado no espaço de estados, e não expandimos um estado se já o visitámos pelo menos k vezes, onde k é um inteiro maior do que 1 (quando k=1 este algoritmo é equivalente ao algoritmo de procura em profundidade em grafos standard). Para cada uma das seguintes frases, indique se é verdadeira (V) ou falsa (F). Cada resposta correta vale 0,4val, cada resposta errada vale -0,4val. A pergunta tem uma cotação mínima de 0 valores.

- O novo algoritmo é completo, enquanto a versão standard da procura em profundidade primeiro em grafos não é;
- Se não houver ciclos no espaço de estados, a complexidade temporal do novo algoritmo é a mesma do que a do algoritmo de procura em profundidade primeiro em árvores;
- A complexidade espacial do novo algoritmo é a mesma do que a do algoritmo de procura em profundidade primeiro em árvores;
- A complexidade espacial do novo algoritmo é a mesma do que a do algoritmo de procura em largura primeiro em árvores;
- Nenhuma das frases anteriores é verdadeira.

IV) [2val] Considere uma equipa com P Pacmen, indistinguíveis entre si, que têm como objetivo comer todas as maçãs existentes numa grelha de tamanho MxN. Para comer uma maçã, basta que um Pacman se mova para a célula onde a maçã se encontra. Em cada momento, cada um dos Pacmen pode escolher uma das seguintes quatro ações: esquerda, direita, para cima, para baixo, mas não podem colidir entre si. Por outras palavras, qualquer ação que resulte em dois ou mais Pacmen ocuparem o mesmo quadrado não resultará em qualquer movimento para cada um deles. Há um total de G maçãs, que são indistinguíveis umas das outras. Tratando este cenário como um problema de procura, consideramos cada configuração da grelha como um estado, e o estado objetivo como uma configuração onde não há maçãs na grelha. Independente do número de Pacmen que se movam, o custo total do movimento é 1. As figuras ilustram um exemplo de um estado inicial bem como um exemplo de um estado objetivo.



- a) Assuma que a equipa tem apenas um Pacman i.e., $P=1$.
 - i) Qual o tamanho da menor representação do espaço de estados?
 - ii) Para cada uma das seguintes heurísticas, indique se é apenas consistente (C), apenas admissível (A), ambas (CA) ou nenhuma (N);
 - (1) Soma das distâncias de Manhattan do Pacman a cada maçã;
 - (2) Número de maçãs vezes a distância de Manhattan máxima do Pacman a uma maçã;
 - (3) Número de maçãs restantes.
- b) Assuma agora que existem tantos Pacmen como existem maçãs i.e., $P=G$. Recorde-se que os vários Pacmen são indistinguíveis.
 - i) Qual o tamanho da menor representação do espaço de estados?
 - ii) Para cada uma das seguintes heurísticas, indique se é apenas consistente (C), apenas admissível (A), ambas (CA) ou nenhuma (N);
 - (1) A maior das distâncias de Manhattan entre cada Pacman e a maçã que lhe está mais próxima;
 - (2) A menor das distâncias de Manhattan entre cada Pacman e a maçã que lhe está mais próxima;
 - (3) Número de maçãs restantes.
 - (4) Número de maçãs restantes dividido por P.

XX

V) [3val] Nesta pergunta, vamos usar o algoritmo DPLL para verificar se a seguinte frase S é ou não satisfazível:

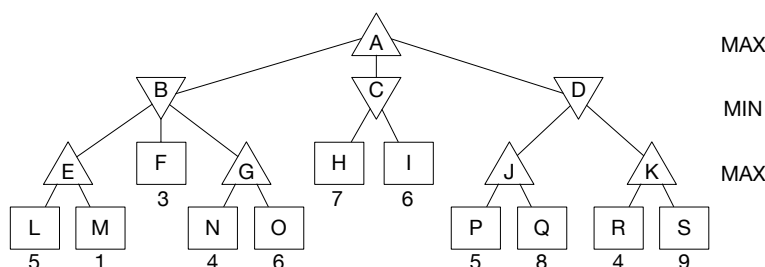
$$S: (A \vee \neg B \vee D) \wedge (\neg A \vee B \vee E) \wedge (A \vee \neg C \vee \neg F) \wedge (B \vee C \vee F) \wedge (\neg C \vee \neg D \vee \neg E) \wedge (D \vee E \vee F)$$

Durante a execução do algoritmo, imagine que já fez as seguintes atribuições: A=Falso e B=Verdadeiro.

- a) Que variável é agora um símbolo puro?
- b) Que valor de verdade se lhe deve atribuir?
- c) Qual das clausulas originais é agora uma clausula unitária?

i. $(A \vee \neg B \vee D)$	iii. $(A \vee \neg C \vee \neg F)$	v. $(\neg C \vee \neg D \vee \neg E)$
ii. $(\neg A \vee B \vee E)$	iv. $(B \vee C \vee F)$	vi. $(D \vee E \vee F)$
- d) O que devemos fazer depois de encontrar esta clausula unitária?
 - i. Terminar, dado que a frase S é insatisfazível;
 - ii. Terminar, devolvendo a atribuição que satisfaz a frase S;
 - iii. Satisfazer a clausula unitária atribuindo o valor Verdadeiro às restantes variáveis nessa clausula;
 - iv. Satisfazer a clausula unitária atribuindo o valor Falso às restantes variáveis nessa clausula.
- e) Existe alguma atribuição às variáveis que satisfaça S onde A=Falso e B=Verdadeiro?
- f) Ao resolver este problema, reparou que resolver o símbolo puro não acrescentou mais clausulas unitárias. É possível que a resolução de um literal puro alguma vez acrescente uma nova clausula unitária (não necessariamente neste exemplo)? Justifique brevemente.

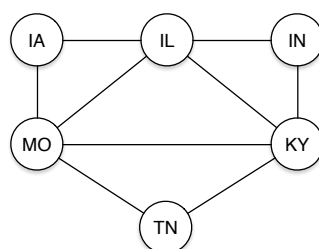
VI) [3val] Considere a árvore de jogo de dois jogadores (MAX e MIN), onde os valores nas folhas são estimativas do ganho para MAX a partir desse estado e os filhos de um nó são visitados da esquerda para a direita.



- Calcule o valor MINIMAX de cada nó não terminal e indique qual o movimento que MAX deverá escolher.
- Assumindo que a árvore é percorrida da esquerda para a direita, indique quais os nós que nunca chegariam a ser visitados pelo algoritmo de procura α - β .
- Se os ramos de cada nó MAX forem reordenados de modo a maximizar o número de folhas que nunca chegariam a ser visitadas pelo algoritmo de procura α - β , qual seria o número de **folhas** que não seriam visitadas?

XX

VII) [3val] Considere o seguinte grafo de restrições, representando parte do mapa dos Estados Unidos que pretendemos colorir com três cores (R)osa, (G)rená e (B)ege, de modo a que estados adjacentes, ligados por um arco, não tenham a mesma cor.



IA - Iowa
IL - Illinois
IN - Indiana
MO - Missouri
KY - Kentucky
TN - Tennessee

- Assumindo que à variável TN já foi atribuída a cor Rosa, aplique o algoritmo de Verificação para a Frente (Forward Checking) e elimine (na folha de resposta) as cores que seriam descartadas para as restantes variáveis.
- Assumindo que à variável TN já foi atribuída a cor Rosa, à variável MO a cor Grená e que não foi efetuada Verificação para a Frente (Forward Checking), aplique o algoritmo de Consistência de Arcos (AC-3) e elimine (na folha de resposta) as cores que seriam descartadas para as restantes variáveis.
- Considere os seguintes domínios possíveis, obtidos após seleção da variável IA e atribuição da cor Grená, seguida da propagação de restrições. Das restantes variáveis, quais poderiam ser selecionadas pela heurística da variável mais constrangida para serem consideradas a seguir?

TN	IN	IA	IL	MO	KY
R G B	R G B	G	R B	R B	R G B

- Considere os seguintes domínios possíveis, obtidos após seleção da variável IA e atribuição da cor Grená, seguida da propagação de restrições (o mesmo da alínea anterior). Ignore a heurística da variável mais constrangida. Das restantes variáveis, quais poderiam ser selecionadas pela heurística da variável mais constrangedora para serem consideradas a seguir?

TN	IN	IA	IL	MO	KY
R G B	R G B	G	R B	R B	R G B

- Considere a seguinte atribuição completa, mas inconsistente. A variável IA foi selecionada para lhe ser atribuído um novo valor durante a execução de um algoritmo de procura local para encontrar uma atribuição completa e consistente. Que valor seria atribuído à variável IA de acordo com a heurística min-conflitos?

TN	IN	IA	IL	MO	KY
B	G	?	G	B	B

- Ignore as alíneas anteriores. Encontre uma solução para o problema aplicando o algoritmo de procura com retrocesso (backtrack) em que é usada a heurística da variável mais constrangida e desempate pela heurística da variável mais constrangedora (e em caso de persistência de empate, por ordem alfabética crescente), e após cada atribuição de um valor a uma variável é executado o algoritmo de consistência de arcos (AC3) para a redução dos domínios das variáveis. Os valores (cores) devem ser atribuídos por ordem alfabética crescente. Qual a solução encontrada? Quantas vezes teve de retroceder (backtrack)?

VIII) [Bónus: até 2val] Para cada alínea, indique se ela é verdadeira (V) ou falsa (F). Cada resposta correta vale 0,4val, cada resposta errada vale -0,4val. A pergunta tem uma cotação mínima de 0 valores.

- a) Considere um problema de procura no qual o espaço de estados inclui a variável x . Então, quaisquer quantidades que sejam uma função determinística de x não têm que ser incluídas no espaço de estados.
- b) Se fixarmos a temperatura T para ser igual a 0, o algoritmo de recristalização simulada (*simulated annealing*) comporta-se como o trepa-colinas standard.
- c) A complexidade temporal de um algoritmo eficiente para resolver problemas de satisfação de restrições arbóreos é linear no número de variáveis.
- d) Seja $h_1(s)$ uma heurística consistente e $h_2(s)$ uma heurística admissível, então $h_3(s) = \min(h_1(s), h_2(s))$ é necessariamente consistente.
- e) Um problema de satisfação de restrições apenas com variáveis booleanas pode sempre ser resolvido em tempo polinomial.

FIM!

Nome: _____

Número: _____

D)	
i	S B D E C F
ii	S B C
iii	S S B C S B D E C F
iv	S C F
v	S B E D C
vi	S C

II)	G1	G2	G3
i	X	X	X
ii	X		X
iii		X	
iv	X	X	X
v	X		X
vi		X	
vii		X	

III) [V/F]	
i	F
ii	F
iii	F
iv	F
v	T

IV.a.i)	a.ii.1)	a.ii.2)	a.ii.3)	b.i)	b.ii.1)	b.ii.2)	b.ii.3)	b.ii.4)
$MN(2)^G$	N	N	CA	$\binom{MN}{P} 2^G$	N	CA	N	CA

V.a)	b) [V/F]	c) [i .. vi]	d) [i .. iv]	e) [Sim/Não]	f) [Sim/Não]
C	F	i	iii	Sim	Não
<p>f) Justificação: Para produzir uma nova cláusula unitária, seria necessário atribuir um valor de verdade que tornasse falso um literal de uma cláusula, ainda não resolvida, com dois literais por resolver. Resolver um símbolo puro elimina todas as cláusulas em que este aparece, não alterando as restantes, pois esse símbolo puro nelas não aparece. Assim, dado que a resolução de um símbolo puro nunca torna falso um literal existente – apenas torna as suas ocorrências verdadeiras, eliminando as respetivas cláusulas – podemos concluir que nunca irá produzir novas cláusulas unitárias.</p>					

VI	a)	A	B	C	D	E	G	J	K
		8	3	6	8	5	6	8	9
	a) Movim. de MAX:	D	b)	O				c)	5

VII							c)		d)		e)		
a)	TN	IN	IA	IL	MO	KY	IL, MO		KY		R		
	R	R G B	R G B	R G B	■ G B	■ G B	f)	TN	IN	IA	IL	MO	KY
b)	TN	IN	IA	IL	MO	KY		B	R	G	B	R	G
	R	■ G ■	■ ■ B	R ■ ■	G	■ ■ B		f) Número de retrocessos:					0

VIII)	a) [V/F]	V	b) [V/F]	F	c) [V/F]	V	d) [V/F]	F	e) [V/F]	F
-------	----------	---	----------	---	----------	---	----------	---	----------	---