Gabarito proposto

P_1 – Sistemas Inteligentes (INE5633) – 08out2014

Sistemas de Informação – Universidade Federal de Santa Catarina

Estudante: _____

1. *(1,0)* Defina "inteligência artificial" com suas próprias palavras. Detalhe dois exemplos de aplicação.

   Uma, dentre várias possibilidades de respostas: Métodos computacionais para realizar tarefas que, quando tratadas por um ser humano, são consideradas inteligentes (McCarty). Exemplos: processamento de linguagem natural; reconhecimento facial.

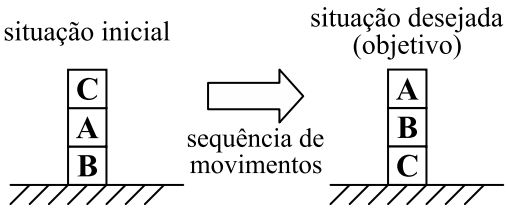
2. *(1,0)* Para cada um dos agentes a seguir, desenvolva uma descrição de PEAS (*Performance*/Medida de desempenho; *Environment*/Ambiente; Atuadores; Sensores) do ambiente de tarefas.

   (a) Agente para definição de rotas entre duas cidades.

   (b) Robô lutador em um campeonato de combate na *Robogames*.

	P	E	A	S
(a)	rotas de; menor caminho	estradas; cidades	ir para cidade vizinha; atualizar distância percorrida	verificar distância; verificar se é destino
(b)	vencer a luta	arena; árbitros; oponente, torcida;	visualizar; empurrar levantar; afastar	detecção de posição; distância ao adversário

3. *(3,0)* Considere o problema *mundo dos blocos* em que três blocos (**A**, **B** e **C**) estão empilhados sobre uma mesa. O problema consiste em uma sequência de movimentos de blocos que, partindo de uma situação inicial, leva a uma situação desejada (objetivo). Veja um exemplo na figura abaixo.



O *Espaço de Estados* deste problema consiste em: situação inicial dos blocos é o estado inicial; a situação desejada é o estado final; e a sequência de movimentos é baseada em uma sequência de operações válidas de transformação de estados. São nove operações possíveis (algumas válidas a cada estado):

- coloque o bloco A: (1) sobre a mesa; ou (2) sobre o bloco B; ou (3) sobre o bloco C
- coloque o bloco B: (4) sobre a mesa; ou (5) sobre o bloco A; ou (6) sobre o bloco C
- coloque o bloco C: (7) sobre a mesa; ou (8) sobre o bloco A; ou (9) sobre o bloco B

Uma operação é válida quando ela respeita as seguintes restrições de movimentos:

(a) é permitido mover apenas um bloco de cada vez;

(b) não pode haver outros blocos em cima do bloco a ser movimentado;

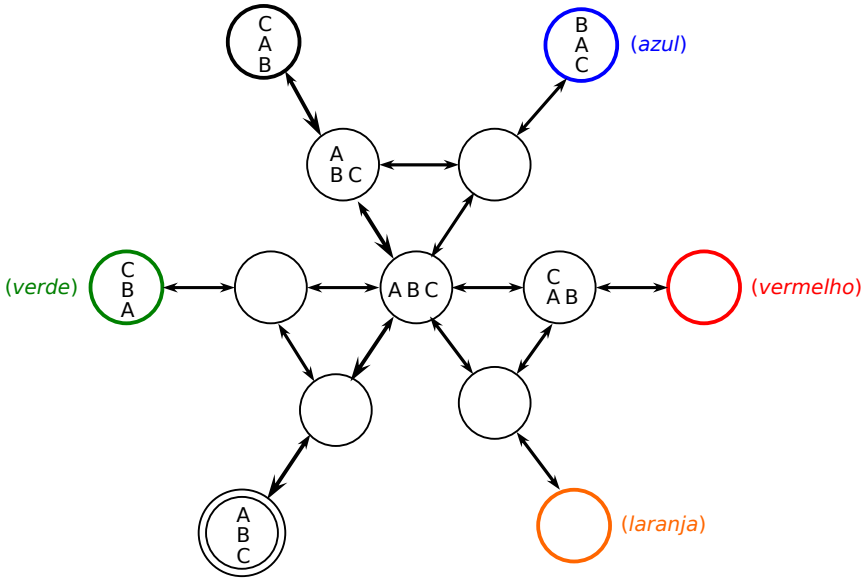
(c) não pode haver outros blocos sobre o qual se quer colocar um bloco em cima.

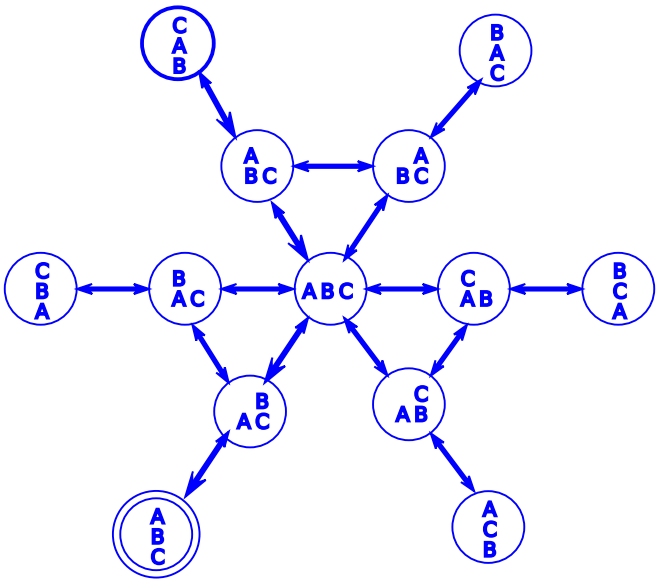
No exemplo da figura, partindo da situação inicial, existe apenas um movimento legal: coloque o bloco **C** sobre a mesa $\underline{\underline{\begin{matrix}A\\B\end{matrix}\quad C}}$. E, na sequência, existem três movimentos legais possíveis:

- ou coloque o bloco **A** sobre a mesa: $\underline{\underline{B\quad C\quad A}}$
- ou coloque o bloco **A** sobre o bloco **C**: $\underline{\underline{\begin{matrix}&A\\B&C\end{matrix}}}$
- ou coloque o bloco **C** sobre o bloco **A**: $\underline{\underline{\begin{matrix}C\\A\\B\end{matrix}}}$

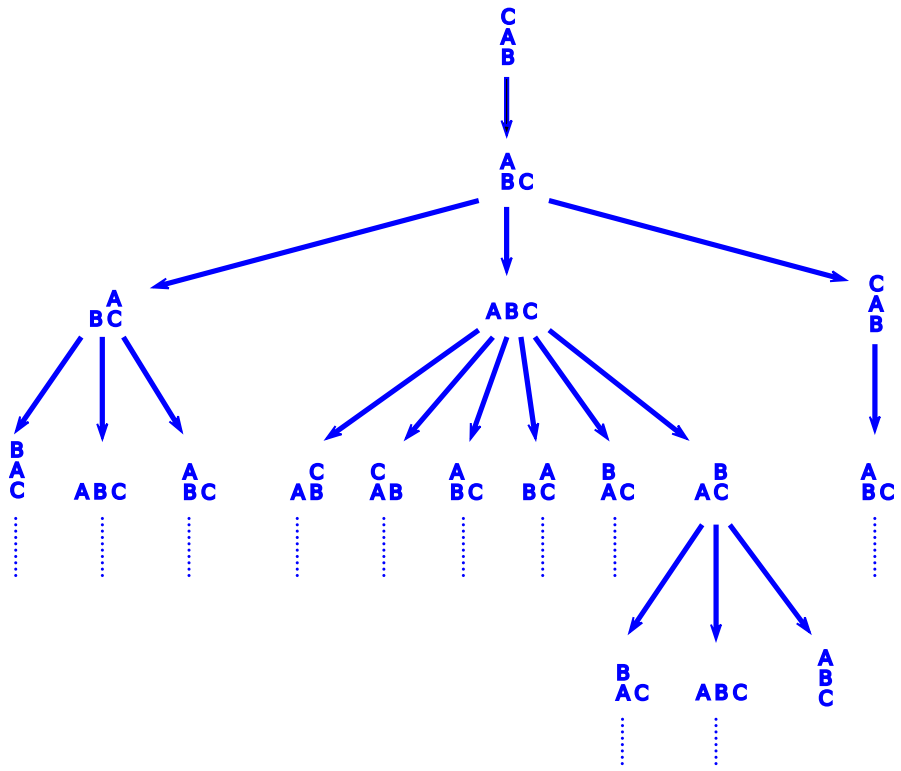
Pede-se:

(a) Complemente o *Espaço de Estados* da figura abaixo (acrescente as letras nos blocos empilhados sobre a mesa com *Estados* vazios).





(b) Selecione um estado inicial (**matrículas com finais 0, 1 e 2**, devem utilizar o verde; **finais 3, 4**, o azul; **finais 5, 6**, o vermelho; **finais 7, 8 e 9**, o laranja), desenhe ao menos quatro níveis (nível da raiz, mais três níveis) da árvore de busca (de pesquisa); e descreva a sequência de visitação dos estados, considerando os seguintes algoritmos:



i. Busca cega (sem informação) com percurso em profundidade (com pilha), evitando repetições.

C A B ; A B C ; A B C ; B A C ; ... ABC ; ...

ii. Busca cega (sem informação) com percurso em largura/extensão (com fila).

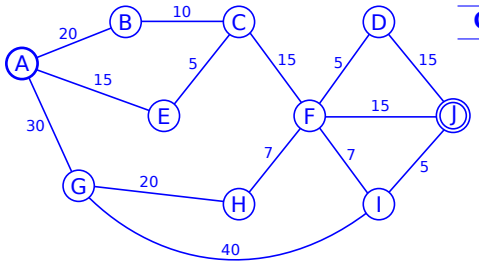
C A B ; A B C ; A B C ; ABC ; C A B ; B A C ; ABC ; A B C ; C A B ;

C A B ; A B C ; A B C ; B A C ; B A C ; A B C ; ...

B A C ; ABC ; A B C

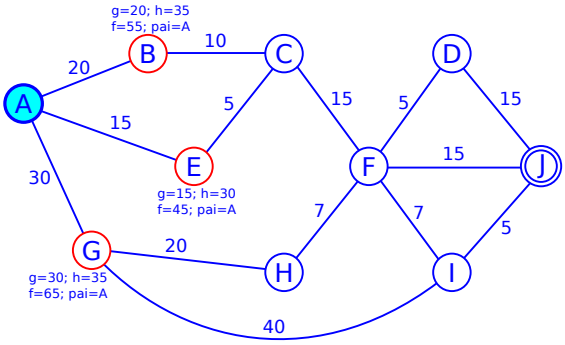
4. *(2,0)* Construa um grafo com, no mínimo, sete nós, para representar cidades (nomeadas por A, B, C, ...) e, no mínimo, dez arestas para representar as rodovias de ligação. Considere que há cidades em que partem/chegam ao menos três rodovias. Atribua, a cada aresta, uma quilometragem. Nesta estrutura, escolha uma cidade de origem e outra de destino (de forma que o caminho mínimo entre elas passe por

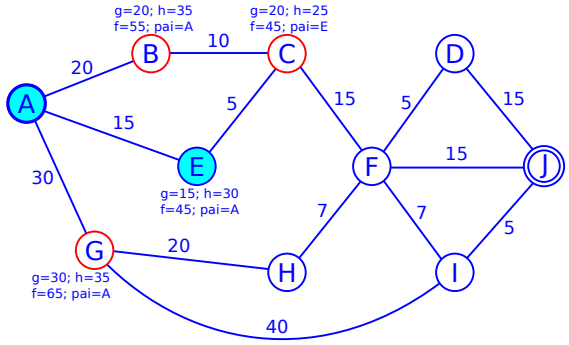
ao menos duas outras cidades), e efetue busca com informação, utilizando o algoritmo A*. Deixe clara a heurística adotada e sua tabela de valores de $h(n)$ para cada cidade n.

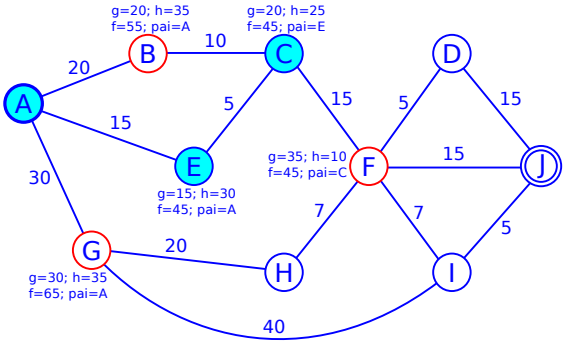


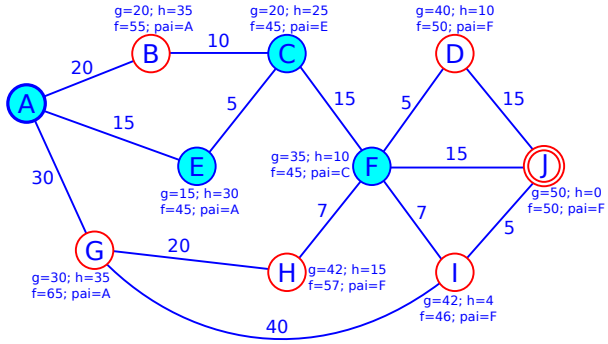
Cidade n	**Heurística** $h(n)$	
A	40	← origem
B	35	
C	25	
D	10	
E	30	
F	10	
G	35	
H	15	
I	4	
J	0	← destino

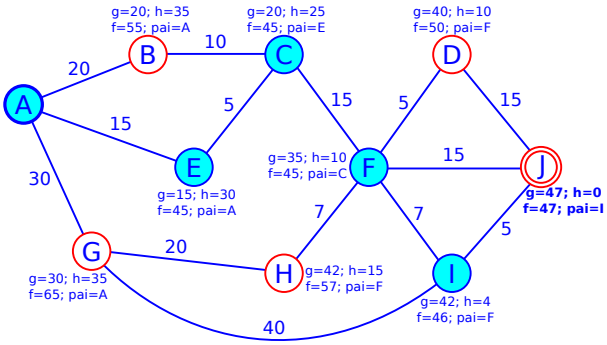
A cidade origem escolhida é A e a destino é J. A heurística adotada é a distância Euclidiana (menor em linha reta) de cada cidade à J (veja tabela ao lado). Segue busca com A*:
$A \rightarrow E$ $(15km)$, $E \rightarrow C$ $(20km)$, $C \rightarrow F$ $(35km)$, $F \rightarrow I$ $(42km)$, $I \rightarrow J$ $(47km)$

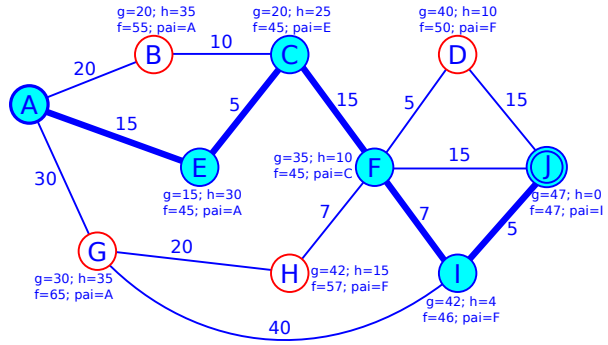










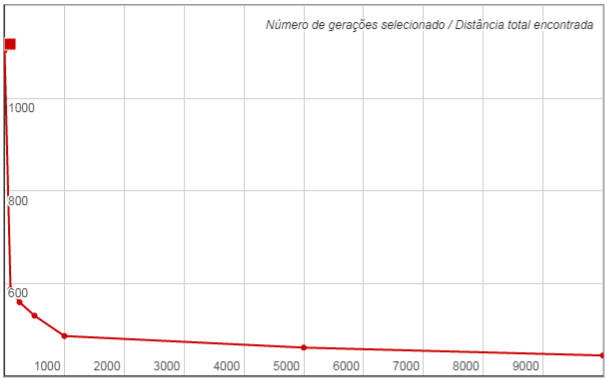


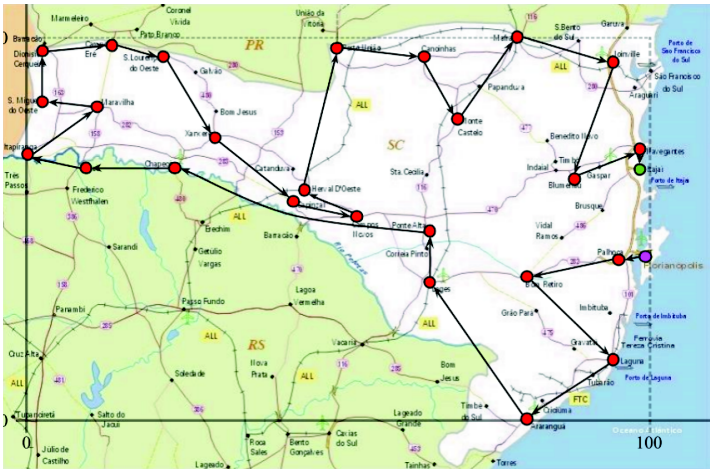
5. *(1,0)* Selecione e explique **apenas um** dos algoritmos de busca local listados a seguir, e apresente um exemplo de problema de otimização associado ao método.

   - Busca de subida de encosta (*hill climbing*).
     O algoritmo consiste em uma repetição que percorre o espaço de estados no sentido do valor crescente (ou decrescente). Termina quando encontra um pico (ou vale) em que nenhuma vizinho tem valor mais alto. Não mantém uma árvore, o nó atual só registra o estado atual e o valor da função objetivo. Não examina antecipadamente valores de estados além dos valores dos vizinhos imediatos do estado atual. Exemplo: Problema das 8-rainhas.
   - Têmpera simulada (*simulated anealing*).
     Combina a subida de encosta com um percurso aleatório resultando em eficiência e completeza. Subida de encosta dando uma "chacoalhada" nos estados sucessores (estados com avaliação pior podem ser escolhidos com uma certa probabilidade; esta probabilidade diminui com o tempo). Exemplos: projetos de circuitos integrados, layout de instalações industriais, otimização de redes de telecomunicações.
   - Feixe local (*local beam*).
     Manter k estados em vez de um. Começa com k estados gerados aleatoriamente. A cada iteração, todos os sucessores dos k estados são gerados. Se qualquer um deles for o estado objetivo, a busca para; se não seleciona-se os k melhores estados da lista pra continuar. Exemplo: mesmos problemas de busca anteriores.
   - Algoritmo genético (*genetic algorithm*).
     Um estado sucessor é gerado por meio da combinação de dois estados pais. Começa com k estados gerados aleatoriamente (população). Um estado é representado por um string de um alfabeto finito (normalmente strings de 0s e 1s). Função de avaliação (função de fitness). Valores mais altos pra estados melhores. Produz a próxima geração de estados por seleção, mutação e crossover. Exemplo: Problema da mochila.

6. *(2,0)* **A ser entregue pelo Moodle até às 10h de amanhã, 09/10.** Pesquise e explique o PCV-Problema do Caixeiro Viajante (ou *TSP-Travelling Salesman Problem*) utilizando algoritmo genético (meta-heurístico). Aplique-o para a visitação de todas as cidades catarinenses do trabalho 1, faça um gráfico da distância total encontrada em função do número de gerações selecionado (faça ao menos 5 testes diferentes), e desenhe (sobre o mapa) o percurso final do melhor resultado encontrado. São dados:

   - Projeto em Java[1] (Netbeans): http://www.inf.ufsc.br/∼alexandre.silva/ine5633/provas/tspGA.zip
   - Coordenadas normalizadas: http://www.inf.ufsc.br/∼alexandre.silva/ine5633/provas/tspXY.pdf





[1] Extraído de *The Project Spot*.