

# Primeira Avaliação

## Planejamento de caminhos

### 1 Descrição

A **Robótica** vem ganhando notoriedade nas últimas décadas, fazendo-se presente na indústria, nos atendimentos automáticos e até dentro de nossas casas. Uma área que tem se destacado e obtido grandes avanços em anos mais recentes é a **robótica móvel**, responsável por estudar robôs capazes de locomoverem-se no ambiente em que estão inseridos. Nesse contexto, o **planejamento de caminhos** consiste em determinar quais pontos do ambiente devem ser visitados por um robô para que ele saia de um ponto de origem e alcance um ponto de destino, evitando colisões com obstáculos presentes no ambiente.

Este trabalho tem como objetivo o planejamento de caminhos de menor custo e livre de colisões, que permitam que um robô, a partir de sua posição inicial, alcance as saídas de uma arena. Nas seções a seguir são descritas com detalhes as informações necessárias para o desenvolvimento do seu programa.

#### 1.1 As arenas

As arenas consideradas nesse trabalho possuem o formato retangular e são divididas em elementos menores, chamados de células, como podemos ver na Figura 1.

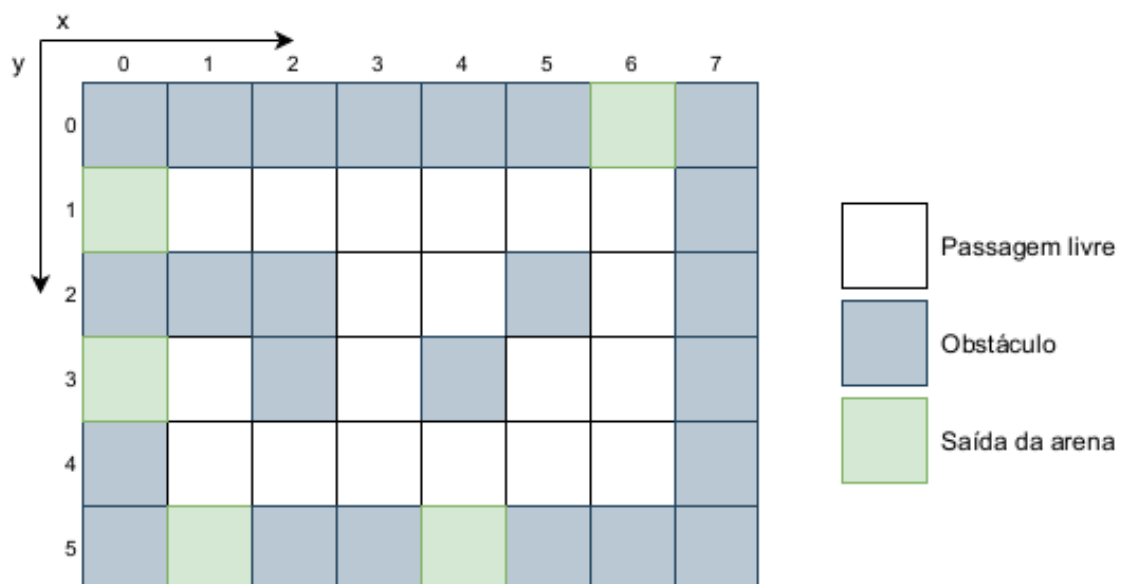


Figura 1: Ilustração de uma arena com 48 células e cinco saídas.

Cada célula da arena é um quadrado de  $1m \times 1m$  e é identificada por um par de inteiros  $(x, y)$ , que representam as coordenadas da célula dentro da arena. Por

exemplo, na Figura 1, temos que o par  $(0, 0)$  identifica a célula superior esquerda, enquanto o par  $(7, 5)$  identifica a célula inferior direita da arena. Podemos notar que todas as células que estão na borda da arena representam um obstáculo ou uma saída. Um obstáculo, além de aparecer nas bordas, também pode ocorrer em qualquer outra posição, como na célula  $(4, 3)$  da arena da Figura 1, por exemplo. Já as células que representam uma saída da arena podem estar localizadas apenas nas bordas. Uma arena pode não ter saídas, ou possuir de uma até dez saídas.

## 1.2 O robô

O robô para o qual você deve planejar os caminhos é controlado por comandos simples e possui um tempo máximo  $t$  de segundos para se locomover pela arena. Uma vez que o robô está posicionado na célula  $(x, y)$  da arena ele pode receber os seguintes comandos:

- **c** - faz o robô se locomover 1 metro para cima (em direção ao  $y == 0$ );
- **d** - faz o robô se locomover 1 metro para direita (em direção oposta ao  $x == 0$ );
- **b** - faz o robô se locomover 1 metro para baixo (em direção oposta ao  $y == 0$ );
- **e** - faz o robô se locomover 1 metro para esquerda (em direção ao  $x == 0$ ).

Esses comandos fazem com que o robô consiga se locomover apenas entre células vizinhas e apenas em quatro direções, como ilustra a Figura 2.

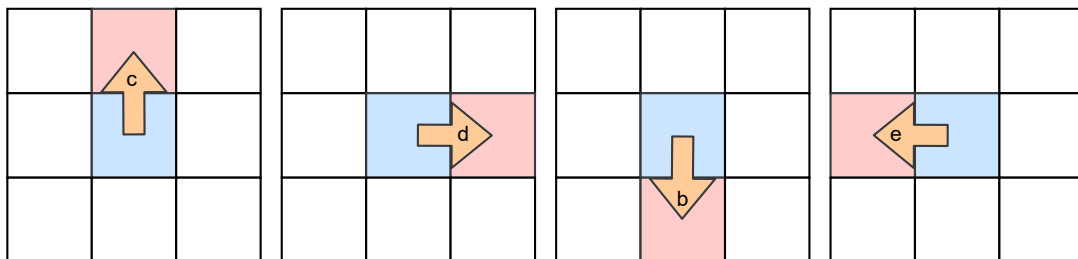


Figura 2: Ilustração dos comandos que o robô pode receber. Inicialmente o robô encontra-se na célula azul e, após o comando indicado pela seta, ele alcança a célula vermelha.

O robô se movimenta, para qualquer uma das direções permitidas, a uma velocidade constante de 1 metro por segundo. O tempo de movimentação do robô, ou a distância que ele consegue percorrer, é limitado pela duração da carga de sua bateria de lítio. Isso quer dizer que, dependendo da posição do robô, ele pode não alcançar algumas células da arena.

Na próxima seção será detalhado o que você deve implementar e são dados alguns exemplos de entrada e saída para o problema.

## 2 Implementação

A implementação desse trabalho consiste em ler informações do estado de uma arena e em planejar os caminhos mínimos para que o robô alcance as saídas da arena caso seja possível. A entrada do seu programa é feita através de um arquivo de configuração. Esse arquivo **deve** ser informado para seu programa via argumento de programa. Ou seja, o nome completo do arquivo **deve** ser passado para seu programa antes do início da execução. Por exemplo, suponha que o arquivo de configuração esteja salvo com o nome `arena.dat`, e que seu executável esteja nomeado como `simulacao`. A chamada de execução do seu programa **deve** ser realizada da seguinte maneira:

```
user@alg2:~$./simulacao arena.dat
```

O arquivo de configuração é um arquivo de texto, como o mostrado na Figura 3. A primeira linha desse arquivo traz a identificação da instância de teste, enquanto as duas próximas linhas indicam, respectivamente, o tempo máximo que o robô pode se locomover e a quantidade de saídas existentes. A quarta linha do arquivo apresenta as dimensões da arena e, a partir da quinta linha, temos a descrição do conteúdo de cada célula com os possíveis valores:

- ' ' - indica uma célula com passagem livre;
- '#' - indica uma célula com um obstáculo;
- '\*' - indica a posição inicial do robô;
- '0' - '9' - valores entre '0' e '9', inclusive, indicam as saídas da arena. Por exemplo, se uma arena possui 3 saídas, os valores '0', '1' e '2' irão aparecer nas bordas da arena indicando tais saídas.

1	Instância ExemploArena	#	#	#	#	#	0	#
2	Orçamento: 7	1						#
3	Saídas: 5	#	#	#			#	#
4	Dimensões: 6x8	2		#		#		#
5	#####0#	#					*	#
6	1 #	#	3	#	#	4	#	#
7	### # #							
8	2 # # #							
9	# *#							
10	#3##4###							

Figura 3: Conteúdo do arquivo de configuração (lado esquerdo) que representa uma arena (lado direito).

Note que as dimensões indicadas no arquivo de configuração são dadas pensando em uma representação matricial da arena (linhas  $\times$  colunas). As dimensões da arena devem ser limitadas exclusivamente pela memória livre do computador. Em outras palavras, o espaço utilizado para armazenar a arena na memória deve ser alocado durante a execução do seu programa.

## 2.1 O que calcular?

Uma vez que é conhecida a entrada, seu programa deve calcular, caso existam, as listas de comandos que devem ser passados para o robô para que ele saia de sua posição inicial e realize o menor caminho que alcance cada uma das saídas da arena, respeitando o limite de tempo estabelecido. Além disso, seu programa deve indicar qual o caminho mais rápido que o robô deve seguir para sair da arena. Lembre que os comandos passados para o robô são **cima**, **direita**, **baixo** ou **esquerda**.

A saída do seu programa deverá ser feita de duas formas: no terminal e em um arquivo. No terminal serão indicadas apenas as saídas mais próximas que o robô consegue alcançar, enquanto no arquivo de resposta estarão os tempos necessários e os comandos que levam o robô para todas as saídas da arena. O nome do seu arquivo de saída deverá ser o nome da instância de entrada, indicada no arquivo de configuração, concatenado com a extensão “.out”.

Na Figura 4 são mostradas as saídas esperadas para o arquivo de entrada ilustrado na Figura 3. Note que o robô não consegue alcançar a saída 1 com o tempo estipulado para ele. Além disso, na Figura 4 é mostrado no terminal apenas o caminho que leva para a saída 4, que é a mais próxima do robô. No entanto, caso existissem mais saídas que fossem alcançadas com o mesmo tempo, todas as saídas e os comandos para alcançá-las deveriam ser listadas ordenadas pelo identificador das saídas.

Instância ExemploArena Orçamento disponível: 7 segundos Quantidade de saídas alcançadas: 4 Saída(s) mais próxima(s): 3 segundo(s)  Saída 4: eeb  Arquivo ExemploArena.out gerado com sucesso!	Posição inicial [x, y]: [6, 4] Orçamento disponível: 7 segundos  Saída 0: 4 segundos -> cccc Saída 1: Não alcançada! Saída 2: 7 segundos -> eeeeece Saída 3: 6 segundos -> eeeeeb Saída 4: 3 segundos -> eeb
--	---

Figura 4: Exemplo de saída apresentada no terminal (esquerda) e saída salva no arquivo (direita).

### Algumas considerações importantes:

- Existem muitas maneiras de calcular os caminhos desejados. Encontre a que seja mais fácil para você e que você realmente entenda o que está fazendo. Nesse trabalho não estamos interessados na eficiência dos algoritmos, por enquanto.



- As configurações das arenas e a posição inicial do robô são sempre válidas.
- Você vai notar que podem existir mais de um caminho, com o mesmo tempo, que leva o robô à mesma saída. Nesses casos, apenas um dos caminhos deve ser listado na tela e salvo no arquivo.
  - Atenção: esse item se refere a caminhos diferentes, com o mesmo tempo, que levam à mesma saída. Isso é diferente do que foi descrito anteriormente, onde caminhos com o mesmo tempo mínimo, que levam à saídas diferentes, devem ser listado na tela.
- A Figura 5 ilustra como seu programa deve responder nos casos onde as arenas não possuem saída.

Instância ExemploArena Orçamento disponível: 15 segundos Sem saída!	Posição inicial [x, y]: [6, 4] Orçamento disponível: 15 segundos  Arena sem saída!
Arquivo ExemploArena.out gerado com sucesso!	

Figura 5: Exemplo de saída apresentada no terminal (esquerda) e saída salva no arquivo (direita) quando a arena não possui saída.

- A Figura 6 ilustra como seu programa deve responder nos casos onde as arenas possuem saídas, mas o robô não consegue alcançá-las devido a falta de tempo ou por elas estarem bloqueadas.

Instância ExemploArena Orçamento disponível: 0 segundos Sem saída!	Posição inicial [x, y]: [6, 4] Orçamento disponível: 0 segundos  Saída 0: Não alcançada! Saída 1: Não alcançada! Saída 2: Não alcançada! Saída 3: Não alcançada! Saída 4: Não alcançada!
Arquivo ExemploArena.out gerado com sucesso!	

Figura 6: Exemplo de saída apresentada no terminal (esquerda) e saída salva no arquivo (direita) quando o robô não consegue alcançar nenhuma saída da arena.

### 3 Relatório

Além do programa que planeja os caminhos, como detalhado na seção anterior, você também deverá entregar um relatório da implementação. Este trabalho poderá ser realizado em grupos, porém os relatórios devem ser feitos individualmente. Esperamos que você descreva no seu relatório a sua solução em alto nível, ou seja,

descreva os principais passos da sua solução sem entrar no nível de código. É interessante também que o relatório informe como estão organizados os dados do seu programa e as principais estruturas de dados (pilha, fila, lista, etc.) que você utilizou, e como as utilizou, na sua solução.

Pense no relatório como uma chance de explicar para os professores as decisões que você tomou durante a implementação do seu programa e os problemas que foram enfrentados nesse período. Por isso, é importante também que você informe as principais dificuldades encontradas e as maneiras adotadas para tentar superar tais dificuldades. O seu relatório deve demonstrar que você tem domínio sobre o que foi desenvolvido, mesmo que, por algum motivo, não tenha obtido o sucesso desejado.

Ao final do relatório, faça uma auto-avaliação sobre sua participação no desenvolvimento do trabalho. Finalize a auto-avaliação com uma nota entre zero e dez, inclusive. Em caso de trabalhos realizados em grupo, faça também uma breve avaliação dos seus colegas, incluindo também uma nota para eles. Fique tranquilo, as informações colocadas nos relatórios não serão divulgadas.

O relatório tem um peso importante na nota final do trabalho. Seja organizado e claro na elaboração, sempre tomando cuidado com erros gramaticais e de concordância.

## 4 Entrega

Instruções para entrega do seu trabalho:

### 1. Codificação e execução

O código-fonte entregue, será compilado com os seguintes parâmetros:

```
gcc avall.c -o avall -ansi -pedantic -Wall
```

Caso o código-fonte não compile com este comando, o trabalho não será considerado e receberá nota **ZERO**. É permitido o uso das seguintes bibliotecas:

- `stdio.h`
- `stdlib.h`
- `string.h`

É proibido o uso de qualquer outra biblioteca. Caso utilize bibliotecas não permitidas, o trabalho não será considerado e receberá nota **ZERO**.

### 2. Formação de grupos

O trabalho pode ser realizado em grupos com, no máximo, dois integrantes (duplas). Porém, os relatórios devem ser **INDIVIDUAIS**. Por mais que a implementação tenha sido feita em conjunto, as dificuldades encontradas e o modo de descrever a solução em alto nível muda de pessoa para pessoa. Portanto, evite cópias nos relatórios, mesmo que parciais. Relatórios de um trabalho que sejam considerados muito semelhantes terão penalidades na nota atribuída.



### 3. Forma de Entrega

A entrega deverá ser realizada no Ambiente Virtual de Aprendizagem da UFMS (AVA) **até as 23:59 do dia 07 de outubro de 2021**. Serão disponibilizados dois links para entrega das atividades: um para o código-fonte na linguagem C e outro para os relatórios. Caso o trabalho seja realizado em grupo, apenas um dos integrantes deve encaminhar o código-fonte no primeiro link.

Encerrado o prazo, não serão mais aceitos trabalhos. Portanto, não deixe para entregar seu trabalho na última hora. Para prevenir imprevistos como queda de energia, problemas com o sistema, falha de conexão com a internet, sugerimos que a entrega do trabalho seja feita pelo menos um dia antes do prazo determinado.

### 4. Arquivo com o programa fonte

Seu arquivo contendo o programa fonte na linguagem C deve estar bem organizado. Um programa na linguagem C tem de ser muito bem compreendido por uma pessoa. Verifique se seu programa tem a indentação adequada, se não tem linhas muito longas, se tem variáveis com nomes significativos, entre outros. Não esqueça que um programa bem descrito e bem organizado é a chave de seu sucesso.

### 5. Conduta ética

Cada grupo tem responsabilidade sobre cópias de seu trabalho, mesmo que parciais. É saudável discutir e consultar colegas de outros grupos, mas sem o compartilhamento do código fonte. Os trabalhos serão submetidos à plataformas de detecção de plágio e, os trabalhos considerados plagiados, mesmo que parcialmente, receberão nota **ZERO**. Ética e respeito ao direito autoral são valores cultivados em nossa profissão.

## 5 Critérios de Avaliação

A nota atribuída a este trabalho será composta por duas notas: programa e relatório. A nota atribuída ao programa levará em consideração as respostas geradas para cada instância de teste e a organização e corretude do código-fonte. Já para a nota do relatório, será considerado, além do uso correto do português, o domínio apresentado pelo acadêmico ao descrever sua solução.

Cada uma das notas descritas acima receberá uma nota entre zero e dez, inclusive. A nota final desse trabalho será dada pela fórmula:

$$A1 = 0.5 \times NP + 0.5 \times NR,$$

onde  $NP$  e  $NR$  são as notas obtidas no programa e no relatório, respectivamente.