

Trabalho Prático 2: Eu, robô

Algoritmos e Estruturas de Dados III – 2016/1

Entrega: 31/05/2016

11 de maio de 2016

1 Introdução

Interessado por pesquisas com robôs, você acabou de entrar em um laboratório de robótica que está participando de um campeonato nacional e, para adquirir experiência, você irá acompanhar a equipe que participará da competição.

Este ano, o problema a ser resolvido, consiste em desenvolver um robô capaz de navegar de forma autônoma e eficiente por uma arena. As equipes receberão um mapa da arena, coordenadas do ponto de origem e o de destino e a equipe vencedora será aquela cujo robô chegar ao destino de forma mais eficiente.

A arena de competição possuirá obstáculos e partes do terreno serão mais difíceis de navegar do que outras. Figura 1 é o mapa de uma das arenas da competição. As células cinzas são obstáculos os quais **o robô não poderá de forma alguma passar por cima**; as células coloridas indicam as regiões por onde o robô pode passar e a intensidade das cores indica a dificuldade da navegação devido às condições do terreno. As células brancas são entradas de uma rede de túneis subterrâneos interconectados, isto é, ao entrar em uma célula branca o robô poderá sair em qualquer outra célula branca no mapa. Estas células são consideradas os atalhos da arena. Para dificultar a competição, os organizadores estabeleceram que em alguns *rounds* os robôs terão seus movimentos restritos de acordo com um par de variáveis d_x e d_y . Por exemplo, se $d_x = 1$ e $d_y = 2$, então para cada iteração de movimento o robô terá que, **obrigatoriamente**, mover uma célula no eixo X e duas células no eixo Y . A direção do movimento (para cima ou para baixo; esquerda ou direita) fica a cargo do software de navegação do robô decidir. **Note que a movimentação em CADA eixo deve ser feita TODA de uma vez, isto é, NÃO é válido intercalar movimentações nos eixos X e Y . Por exemplo,**

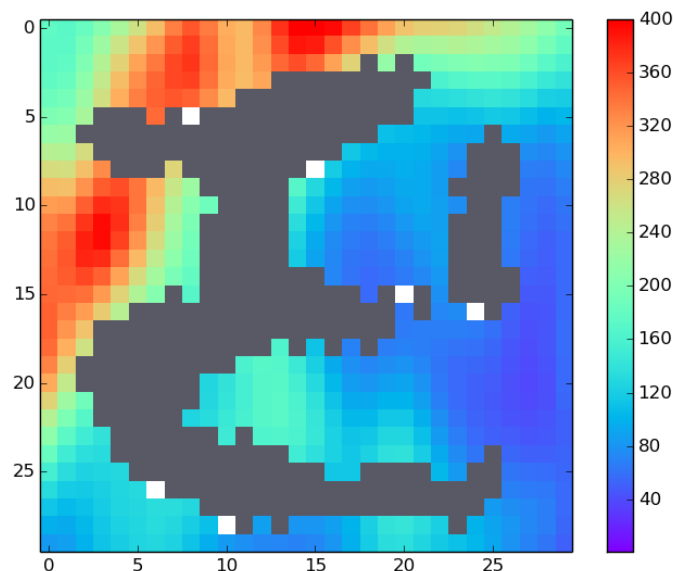


Figura 1: Exemplo de mapa para uma das arenas da competição.

com a restrição anterior é inválido mover uma célula no eixo Y, depois uma em X e finalmente a segunda em Y. Ou o robô move as duas células em Y e depois move em X, ou ele move primeiro em X e depois as duas em Y. Caso $d_x = d_y = 0$, então não há restrição de movimento, isto é para cada iteração o robô move **uma** célula no eixo X **ou** uma célula no eixo Y.

Note que d_x e d_y não terão valor 0, com exceção do caso $d_x = d_y = 0$ descrito acima.

2 O que deve ser implementado

Você deve implementar um programa que, dado um mapa, as restrições de movimento e os pontos de origem, S , e destino, F ; imprima o "custo" mínimo de locomoção de $I \rightarrow F$. Neste caso, custo será o somatório¹ da dificuldade das células por onde o robô andou. Caso não seja possível ir da origem ao destino, o seu programa deverá imprimir -1 .

É obrigatório que este problema seja resolvido como um problema de grafos, utilizando os algoritmos vistos em sala. Além disso, devido à simplicidade do problema, daremos bastante ênfase na eficiência das

¹Veja com atenção o exemplo a seguir para entender como esse cálculo é feito

solução submetidas, tanto avaliação da documentação quanto na execução dos testes. Portanto, utilizem os algoritmos mais eficientes² em termos de complexidade.

O **Makefile** que será submetido com o código deve permitir que o programa seja compilado ao digitar o comando *'make all'* no terminal e o executável deverá ser nomeado: **tp2.exe**. É de extrema importância que esses critérios sejam seguidos à risca, para que seu código seja compatível com os scripts de teste. Caso contrário, o seu TP não poderá ser testado e, consequentemente, você irá tirar 0.

3 Entrada e Saída

A seguir, há um exemplo de como seu programa será executado pelo sistema de teste e do significado de cada um dos parâmetros:

`./tp2.exe mapa.txt s_x s_y f_x f_y d_x d_y`

- s_x : coordenada x do ponto de início
- s_y : coordenada y do ponto de início
- f_x : coordenada x do ponto de término
- f_y : coordenada y do ponto de término
- d_x : restrição de movimento no eixo x
- d_y : restrição de movimento no eixo y

3.1 Entrada

A primeira linha do arquivo conterá dois inteiros M e N , separados por 1 espaço, que indicarão a largura e altura do mapa, respectivamente. Cada uma das N linhas seguintes conterá M **inteiros** separados por espaço, os quais irão descrever as células do mapa:

- -1 : A célula atual é um atalho
- 0 : Obstáculo
- > 0 : Caminho

²Os mais eficientes dentro daqueles que foram ensinados em sala

Note que o valor das células "caminho" corresponde à dificuldade de andar por elas.

Exemplo 1 Considere $I = \langle 0, 0 \rangle$, $F = \langle 1, 2 \rangle$, $d_x = d_y = 0$. Nesse caso o robô irá mover normalmente, uma célula por vez, e o caminho mais eficiente para F é: $P = \{\langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 0, 2 \rangle, \langle 1, 2 \rangle\}$. O custo desse caminho é calculado da seguinte forma. Na primeira iteração o robô move-se de $\langle 0, 0 \rangle$ para $\langle 0, 1 \rangle$ e o custo dessa movimentação é a soma da dificuldade das células envolvidas: $198 + 232 = 430$. Na iteração seguinte, o robô move-se de $\langle 0, 1 \rangle$ para $\langle 0, 2 \rangle$. Logo, o custo dessa locomoção é $232 + 147 = 379$ e o custo total de $\langle 0, 0 \rangle \rightarrow \langle 0, 2 \rangle$ é 809. Note que o custo das células onde o robô se encontra no início da iteração é contabilizado duas vezes. Continuando, para mover-se para $\langle 1, 2 \rangle$ o custo seria 219 e, portanto, o custo total de P será 1028. Agora, imagine que a restrição de movimento deste teste fosse $d_x = 3$ e $d_y = 1$; não haveria nenhum movimento válido. Note que a célula $\langle 3, 1 \rangle$ é a única célula válida e alcançável com a restrição de movimento imposta. Porém, ambos os caminhos que levam a esta célula possuem obstáculos:

- $P_1 = \{\langle 0, 0 \rangle, \langle 1, 0 \rangle, \langle 2, 0 \rangle, \langle 3, 0 \rangle, \langle 3, 1 \rangle\}$
- $P_2 = \{\langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 1 \rangle, \langle 2, 1 \rangle, \langle 3, 1 \rangle\}$

o que é proibido. Portanto, não há nenhum caminho válido de I para F e o seu programa deverá imprimir -1 .

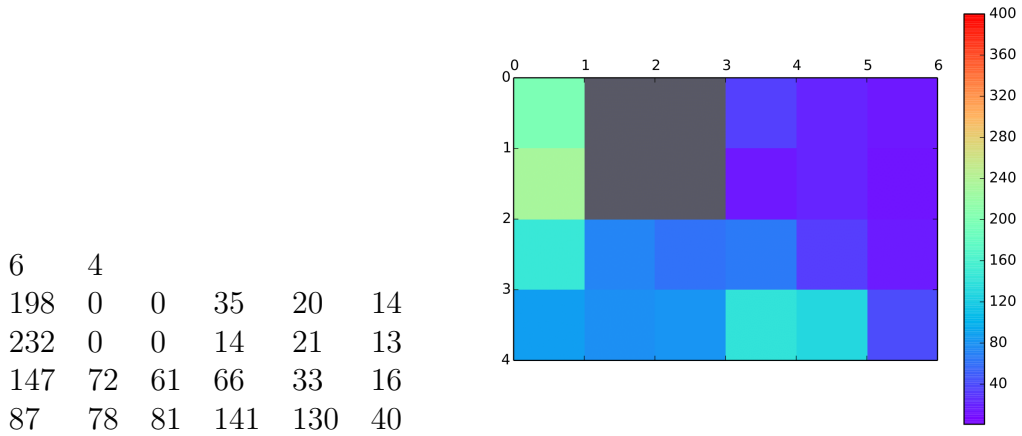


Figura 2: Exemplo de entrada.

Exemplo 2 Considere a seguinte entrada com atalhos nas células $\langle 0, 1 \rangle$, $\langle 1, 3 \rangle$, $\langle 4, 0 \rangle$:

6	4				
198	0	0	35	-1	14
-1	0	0	14	21	13
147	72	61	66	33	16
87	-1	81	141	130	40

Além disso, considere os seguintes parâmetros: $I = \langle 0, 0 \rangle$, $F = \langle 5, 3 \rangle$, $d_x = 1$ e $d_y = 3$. Como primeiro movimento, a única opção do robô é ir de $\langle 0, 0 \rangle$ para $\langle 1, 3 \rangle$ usando o caminho: $P = \{\langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 0, 2 \rangle, \langle 0, 3 \rangle, \langle 1, 3 \rangle\}$ com custo: $C_P = (198 + 0) + (0 + 147) + (147 + 87) + (87 + 0) = 666$. Note que havia um atalho em P , a célula $\langle 0, 1 \rangle$. Porém, como o robô não pode "parar" no meio de um deslocamento ou intercalar deslocamentos em X e Y (Seção 1), ele deve passar direto por esse atalho.

A partir da célula atual $\langle 1, 3 \rangle$ não há nenhum movimento que respeite as restrições de movimentação. Entretanto, esta célula é um atalho e o robô pode utilizá-la para chegar à outra célula atalho $\langle 4, 0 \rangle$ ³ com custo 0. Tomando este caminho, o custo total continua sendo $666 + 0 = 666$. Dada a posição atual, o robô tem como opções utilizar o atalho novamente para atingir qualquer outro atalho no mapa ou movimentar-se de acordo com as restrições e ir para a célula $\langle 3, 3 \rangle$ ou para a célula $\langle 5, 3 \rangle$. Nesse caso, a melhor escolha é ir à $\langle 5, 3 \rangle$ e utilizando o caminhamento $P = \{\langle 4, 0 \rangle, \langle 5, 0 \rangle, \langle 5, 1 \rangle, \langle 5, 2 \rangle, \langle 5, 3 \rangle\}$, de custo $C_P = (0 + 14) + (14 + 13) + (13 + 16) + (16 + 40) = 126$. Logo, o caminho mínimo entre $\langle 0, 0 \rangle \rightarrow \langle 5, 3 \rangle$ tem custo $666 + 0 + 126 = 792$.

Concluindo, o robô só poderá utilizar um atalho se ele estiver em uma célula de atalho antes de **começar** ou depois de **terminar** um movimento, onde o movimento inclui ambos os eixos.

3.2 Saída

O seu programa deverá escrever na saída padrão o custo do menor caminho $I \rightarrow F$. Caso não seja possível alcançar o destino dado os parâmetros da entrada, o seu programa deverá imprimir -1 . **OBS: Inclua o '\n' após o custo do caminho.**

³Ele poderia ir para a outra célula de atalho $\langle 0, 1 \rangle$, porém essa movimentação leva a um caminho mínimo

4 O que deve ser entregue

Deverá ser submetido um arquivo *.zip* contendo somente uma pasta chamada **tp2** e dentro desta deverá ter: (i) O **pdf** da documentação; (ii) Os arquivos necessários para a execução do programa.

Documentação Poderá ter no máximo 10 páginas e deverá seguir tanto os critérios de avaliação discutidos na Seção 5.1, bem como as diretrizes sobre a elaboração de documentações disponibilizadas no *moodle*. Além desses requisitos básicos, a documentação do tp2 deverá ter:

1. Explicação detalhada da modelagem. Dada a simplicidade do trabalho, daremos grande ênfase na explicação de como você mapeou este problema para um problema de grafos.
2. Análise de complexidade.
3. Experimentos:
 - Mostrar a influência da restrição de movimento no tempo de execução. Explique porque isso ocorre.
 - Avaliar a performance em mapas quantidades variadas de obstáculo. Explique os resultados

Implementação Código fonte do seu TP (*.c* e *.h*) e um arquivo *Makefile* para realizar a compilação do seu código, atendendo aos critérios descritos na Seção 2.

5 Avaliação

Seu trabalho será avaliado de acordo com a qualidade do código e documentação submetidos. Eis uma lista **não exaustiva** dos critérios de avaliação que serão utilizados.

5.1 Documentação

A documentação é uma parte importante do trabalho. Este é o momento em que você **descreve** sua solução de forma **autocontida**. Segue abaixo uma breve descrição do conteúdo que esperamos encontrar na documentação de cada **TP**.

Introdução Inclua uma breve explicação do problema que está sendo resolvido no seu trabalho e um resumo da sua solução.

Solução do Problema Você deve descrever a solução do problema de maneira clara e precisa, detalhando e justificando os algoritmos e estruturas de dados utilizados. Para tal, artifícios como pseudo-códigos, exemplos ou diagramas podem ser úteis. Note que documentar uma solução não é o mesmo que documentar seu código. **Não** é necessário incluir trechos de código em sua documentação nem mostrar detalhes de sua implementação, exceto quando estes influenciem o seu algoritmo principal, o que se torna interessante.

Análise de Complexidade Inclua uma análise de complexidade de tempo e espaço dos principais algoritmos e estrutura de dados utilizados. Cada complexidade apresentada deverá ser devidamente **justificada** para que seja aceita.

Avaliação Experimental Sua documentação deve incluir os resultados de experimentos que avaliem o tempo de execução de seu código em função de características da entrada. Cabe a você gerar entradas para esses experimentos. Por exemplo: se esse trabalho fosse sobre ordenação, seria interessante mostrar como o tempo de execução de cada algoritmo varia quando o número de itens a serem ordenados aumenta. Para tal, um gráfico mostrando o tempo de execução em função do tamanho da entrada pode ser interessante. Você também deve interpretar os resultados obtidos. Comente sobre cada gráfico ou tabela que você apresentar mostrando o que é possível concluir a partir dele. **DICA:** Use o guia e exemplos de documentação disponíveis no moodle.

5.2 Implementação

Linguagem & Ambiente O seu programa deverá ser implementado na linguagem **C** e poderá fazer uso de funções da biblioteca padrão da linguagem. Trabalhos que utilizem qualquer outra linguagem de programação e/ou que façam uso de outras bibliotecas que não a padrão serão zerados. Além disso, certifique-se que seu código compile e funcione corretamente nas máquinas **LINUX** dos laboratórios do DCC.

Casos de teste A sua implementação passará por um processo de correção automatizado e, portanto, o formato da saída do seu programa deve ser idêntico àquele descrito nessa especificação. Saídas com qualquer divergência

serão consideradas erradas, mesmo que as divergências sejam *whitespaces*. e.g. espaços, *tabs*, quebras de linha, etc. Para auxiliá-lo na depuração do seu código, será fornecido um pequeno, **não-exaustivo**, conjunto de entradas e suas respectivas saídas. É seu dever certificar-se que seu código funciona corretamente para qualquer entrada válida.

Alocação Dinâmica Algoritmos e estruturas de dados deverão fazer uso de memória alocada dinamicamente (`malloc()` ou `calloc()`). Certifique-se que seu programa utiliza essas regiões de memória corretamente, pois os monitores penalizarão implementações que realizam *out-of-bounds access* e que tenham vazamento de memória (não desalocar memória dinâmica).

DICA: Utilize `valgrind` antes de submeter o seu TP.

Qualidade do código Seu código também será avaliado no quesito de legibilidade, dando atenção, porém não limitando-se, aos seguintes itens: (i) **INDENTAÇÃO**; (ii) nomes de variável e função descritivos e claros; (iii) Modularização adequada; (iv) Comentários dentro de funções, explicando o que certos trechos mais complicados fazem; (v) Comentários fora de funções, explicando, em alto-nível, o que as funções mais importantes fazem; (vi) funções concisas que desempenham somente uma tarefa; (vii) **Proibido uso de variáveis globais**; (viii) Quem usar `goto` zera a matéria =].

DICA: Consulte o exemplo de código no moodle

6 Considerações Finais

Atrasos Trabalhos poderão ser entregues após o prazo estabelecido, porém sujeitos a uma penalização regida pela seguinte fórmula:

$$\Delta_p = \frac{2^{d-1}}{0.32} \%$$

Por exemplo, se a nota dada pelo corretor for 70 e você entregou o TP com 4 dias corridos de atraso, sua penalização será de $\Delta_p = 25\%$ e, portanto, a sua nota final será: $N_f = 70 \cdot (1 - \Delta_p) = 52.2$. Note que a penalização é exponencial e 6 dias de atraso resultam em uma penalização de 100%.