

Problema A – Labirinto Sem Paredes

O Tiago sempre gostou de quebra-cabeças e desafios de destreza manual. No seu aniversário ele recebeu um dos quebra-cabeças mais complicados do mercado – O Labirinto Sem Paredes.

Este quebra-cabeças consiste numa plataforma quadrangular, que pode ser vista como um quadrado $[0, 1] \times [0, 1]$, com um buraco mais ou menos no meio, com coordenadas Q_x, Q_y , e numa pequena bola de metal, com coordenadas P_x, P_y que queremos que caia no buraco sem sair da plataforma. Os valores de Q_x, Q_y estarão sempre entre 0.15 e 0.85 e os valores de P_x, P_y entre 0 e 1.

Para tal, o Tiago pode inclinar ligeiramente o brinquedo e, devido à sua elevada prática com estes quebra-cabeças, consegue escolher valores x, y e mudar as coordenadas da bola de metal de P_x, P_y para $P_x + x, P_y + y$.

No entanto, há um desafio adicional, o Labirinto Sem Paredes é um quebra-cabeças completamente opaco, pelo que o Tiago não sabe as coordenadas da bola. Felizmente, tendo muito experiência com desafios parecidos, consegue usar os seus ouvidos para ter uma vaga ideia da posição da bola. Nomeadamente, após cada movimento sabe se $P_x \leq Q_x$ e se $P_y \leq Q_y$.

O seu irmão mais novo, o Luís, também quer experimentar o quebra-cabeças, pelo que o Tiago quer ver se o resolve rapidamente. Para tal, pode fazer no máximo 25 movimentos e não pode fazer movimentos demasiado longos, pelo que a soma total dos seus movimentos tem de ser no máximo S .

A tua tarefa é encontrar um algoritmo que permita resolver este desafio, meter a bola de coordenadas iniciais desconhecidas P_x, P_y dentro de um buraco, com coordenadas conhecidas, Q_x, Q_y , dentro dos limites, nomeadamente:

1. São efetuados no máximo 25 movimentos, da forma (x_i, y_i) , com $-1 \leq x_i, y_i \leq 1$;
2. As coordenadas da bola nunca podem sair de $[0, 1] \times [0, 1]$;
3. A distância percorrida pela bola, dada por $\sqrt{x_1^2 + y_1^2} + \sqrt{x_2^2 + y_2^2} + \sqrt{x_3^2 + y_3^2} + \dots$ não pode exceder S .

Consideramos que a bola saiu pelo buraco se as suas coordenadas após todos os movimentos forem

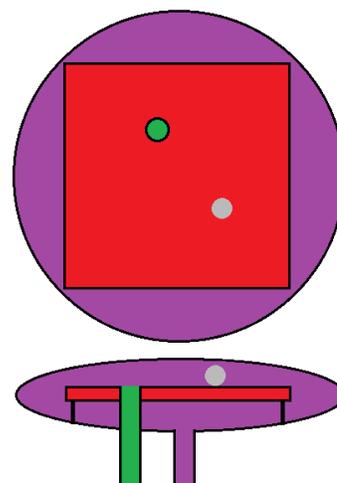


Figura 1: Esquema do quebra-cabeças visto de cima (sem tampa) e o seu corte-transversal lateral. A plataforma está a vermelho, a bola a cinzento, o buraco por onde queremos que a bola caia está a verde. Se a bola sair dos limites do quadrado vermelho, irá cair pelo buraco roxo e o desafio é considerado como perdido e tem de ser recomeçado.

tais que $\sqrt{(P_x - Q_x)^2 + (P_y - Q_y)^2} \leq 10^{-6}$.

Nota que a bola só pode sair pelo buraco após todos os movimentos serem completados. Não se considera que a bola saiu pelo buraco se a bola estiver no buraco e um movimento for efetuado, ou se a bola estiver sobre o buraco a meio de um movimento.

Ficheiros para Download

Podes começar por descarregar os ficheiros correspondentes à tua linguagem (ou um [arquivo zip](#) contendo tudo):

Linguagem	Ficheiro a implementar	Avaliador	Outros Ficheiros	Input exemplo
C++	resolver.cpp	avaliador.cpp	avaliador.h	input.txt

Nota que a implementação do avaliador a usar nos testes oficiais será diferente.

Implementação

Deves submeter um único ficheiro que implementa uma função:

- A função `partida(Qx, Qy, E)`, que recebe as coordenadas do buraco do quebra-cabeças, tal como um inteiro E que codifica se $P_x \leq Q_x$ e se $P_y \leq Q_y$ da seguinte forma: E é 0 se $P_x \leq Q_x$ e $P_y \leq Q_y$, 1 se $P_x > Q_x$ e $P_y \leq Q_y$, 2 se $P_x \leq Q_x$ e $P_y > Q_y$, ou 3 se $P_x > Q_x$ e $P_y > Q_y$.

Para isso deves usar o ficheiro `resolver.cpp` que descarregaste, colocando no interior das funções o teu código. Podes acrescentar outras funções, mas devem ficar todas neste ficheiro que é o único que deves submeter.

Funções a implementar:

C++: `void partida(double Qx, double Qy, int E)`

A tua função deve invocar as seguintes funções:

- A função `mover(x, y)`, que recebe dois números x, y e muda os valores de P_x e P_y para $P_x + x$ e $P_y + y$ respetivamente, e devolve um inteiro E , que codifica se $P_x \leq Q_x$ e se $P_y \leq Q_y$ da forma descrita acima.

Nota o seguinte:

- Os valores de x, y devem estar compreendidos entre -1 e 1 e os valores de P_x, P_y nunca podem estar fora de $[0, 1]$. Se alguma chamada não respeitar estas condições, o teu código terá o resultado de `Wrong Answer`.

- Se excederes 25 perguntas do tipo `mover()` (ou seja, se fizeres mais do que 25 chamadas à função `mover()`), o teu código terá o resultado de **Wrong Answer**.
- Se em algum momento $\sqrt{x_1^2 + y_1^2} + \sqrt{x_2^2 + y_2^2} + \sqrt{x_3^2 + y_3^2} + \dots$ exceder S (seja, a soma das distâncias percorridas em chamadas à função `mover()` exceder S), o teu código terá o resultado de **Wrong Answer**.
- O avaliador oficial irá chamar a função `partida()` várias vezes com diferentes valores. Nota que o avaliador de exemplo só chama esta função uma vez.
- A posição final da bola é dada pelos valores de P_x e P_y no final da execução da função `partida()`.

Funções do avaliador:

C++: `int mover(double x, double y)`

A tua função não deve ler nem escrever para os canais de entrada/saída padrão.

Exemplo

Vamos supor que há $Q_x = Q_y = 0.5$, $P_x = 0.2$ e $P_y = 0.3$. Neste caso o valor inicial de E é 0

Uma possível execução seria a seguinte (com a ordem das chamadas a ser de cima para baixo):

Invocação	Resultado	Descrição	Coordenadas de P
<code>partida(0.5, 0.5, 0)</code>	–	Início	0.2 0.3
<code>mover(0.2, 0.2)</code>	0	Movimento para a direita e para cima	0.4 0.5
<code>mover(0.2, 0.2)</code>	3	Direita e para cima	0.6 0.7
<code>mover(-0.15, -0.1)</code>	2	Esquerda e para baixo	0.45 0.6
<code>mover(0.1, -0.1)</code>	1	Direita e para baixo	0.55, 0.5
<code>mover(-0.1, 0)</code>	0	Para a esquerda	0.45 0.5
<code>mover(0.05, 0)</code>	0	A bola está por cima do buraco	0.5 0.5

Repara que o Tiago poderia ter continuado a mover a bola, possivelmente afastando esta do buraco.

A distância total percorrida neste exemplo foi de $\sqrt{0.2^2 + 0.2^2} + \sqrt{0.2^2 + 0.2^2} + \sqrt{(-0.15)^2 + (-0.1)^2} + \sqrt{0.1^2 + (-0.1)^2} + \sqrt{(-0.1)^2 + 0^2} + \sqrt{0.05^2 + 0^2} \approx 1.03738$

Restrições

São garantidos os seguintes limites em todos os casos de teste desta parte que irão ser colocados ao programa:

$0.15 \leq Q_x, Q_y \leq 0.85$, As coordenadas do buraco
 $0 < P_x, P_y < 1$ As coordenadas iniciais da bola
 $\sqrt{2} \leq S \leq 2$ A distância total que a bola pode percorrer

Sumário de subtarefas

Os casos de teste para este problema estão organizados em quatro grupos:

Grupo	Número de Pontos	Restrições Adicionais
1	25	$Q_x = Q_y = P_x = P_y = 0.5, S = 2$
2	25	$Q_x = Q_y = 0.5, S = 2$
3	25	$S = 2$
4	25	$S = \sqrt{2}$

Testes no vosso computador

É disponibilizado um avaliador exemplo em cada linguagem (`avaliador.cpp`) que pode ser utilizado para testar a tua submissão. Está ainda disponível um ficheiro auxiliar (`avaliador.h`). **Este avaliador não corresponde ao utilizado pelo sistema de avaliação.**

Este avaliador recebe como input cinco números separados por espaços: P_x, P_y, Q_x, Q_y e finalmente S .

O avaliador irá automaticamente invocar a função `partida(Qx, Qy, E)` por ti implementada. Disponibilizamos um ficheiro de teste:

- `input.txt` que contém o caso de exemplo referido acima.

Um exemplo de teste na tua máquina (supondo que tens os compiladores oficiais instalados) seria o seguinte:

Linguagem	Compilar	Executar com o exemplo
C++	<code>g++ -Wall -std=gnu++14 -O2 avaliador.cpp resolver.cpp</code>	<code>./a.out < input.txt</code>

Organização



Alto Patrocínio

Com o Alto Patrocínio de Sua Excelência



O Presidente da República

Patrocinadores



FUNDAÇÃO CALOUSTE GULBENKIAN

Apoios



Prova de Seleção das ONI'2024

Instituto Superior Técnico

Universidade de Lisboa

(8 de Junho de 2024)