

Desafio Tapps

O objetivo deste teste seletivo é identificar o perfil técnico que o candidato apresenta, mostrando uma prévia dos problemas que resolvemos no nosso dia a dia. O teste oferecido a seguir é um desafio que deve ser codificado em uma linguagem de sua preferência, utilizando OO. Os arquivos fonte que compõe a solução devem ser entregues para análise dos técnicos da Tapps.

Os arquivos que compõe a resposta do teste devem ser enviados por e-mail ao contato fornecido pelo departamento de Recursos Humanos. Esses arquivos devem ser entregues na extensão .zip, e um arquivo README dentro do zip deve conter **instruções de compilação e/ou interpretação**, dependendo da tecnologia utilizada na resolução do problema. Em geral, o desafio proposto leva de 2 a 4 horas para ser completado.

Entraremos em contato assim que a resposta tiver sido avaliada pelos nossos entrevistadores técnicos.

Boa sorte!

O Desafio

Considere o seguinte jogo hipotético, que chamaremos de **Bankrupt**, muito semelhante a Banco Imobiliário, onde várias de suas mecânicas foram simplificadas. Numa partida desse jogo, os *players* se alteram em rodadas, numa ordem definida aleatoriamente no começo de cada partida. Os *players* sempre começam uma partida com 300 *coins* cada um.

Nesse jogo, o tabuleiro é composto por 20 propriedades em sequência em uma determinada ordem no tabuleiro. Cada propriedade tem um custo de venda, um valor de aluguel e, caso já estejam compradas, um proprietário. Por simplicidade do problema, no *Bankrupt* **não é possível** construir hotéis e nenhuma outra melhoria sobre as propriedades do tabuleiro.

O valor de venda e de aluguel de todas as 20 propriedades são configurados num arquivo de nome `gameConfig.txt`, que seu programa deve ler e interpretar. Os detalhes sobre a formatação do arquivo se encontram na seção *Entrada*.

No começo da sua vez, o jogador joga um dado equiprovável de 6 faces que determina quantas espaços no tabuleiro o player vai andar.

- Ao cair em uma propriedade sem dono, o jogador pode escolher entre comprar ou não a propriedade. Essa é a única forma pela qual uma propriedade pode ser comprada. Mais detalhes serão explicados à frente.
- Ao cair em uma propriedade que tem dono, o player deverá pagar ao proprietário o valor do aluguel da propriedade.
- Ao completar uma volta no tabuleiro, o jogador ganha 100 *coins*.

Jogadores só podem comprar propriedades caso ela não possua dono e o jogador tenha dinheiro suficiente para a compra. Caso o player esteja em cima de uma propriedade sem dono e não possua *coins* suficientes para executar a compra, ele apenas não a fará. Ao comprar uma propriedade, o jogador perde o dinheiro e ganha a posse da propriedade.

Cada um dos *players* tem uma implementação de comportamento diferente, que dita as ações que eles vão tomar ao longo do jogo. Mais detalhes sobre o comportamento serão explicados à frente.

O jogador que ficar com saldo negativo perde o jogo e não joga mais até o final da partida. Suas propriedades voltam a ficar sem dono e portanto podem ser compradas por qualquer jogador. Para facilitar, o jogador é retirado do jogo no momento em que seu saldo se tornar negativo, não precisando esperar até o fim da rodada. Cuidado para não gerar inflação no jogo (jogador pagando aluguel com dinheiro que não possui).

O *Bankrupt* termina quando restar somente um *player* com dinheiro. Esse *player* é declarado o vencedor.

Desejamos rodar uma simulação sobre o Bankrupt para decidir qual a melhor estratégia. Para isso, idealizamos uma partida com 4 diferentes tipos de possíveis jogadores. Os comportamentos definidos são:

- O jogador 1 é **impulsivo**;
 - Compra qualquer propriedade sobre a qual ele parar.
- O jogador 2 é **exigente**;
 - Compra qualquer propriedade, desde que o aluguel dela seja maior do que 50 *coins*.
- O jogador 3 é **cauteloso**;
 - Compra qualquer propriedade, desde que ao final da compra ele possua uma reserva maior ou igual a 80 *coins*.
- O jogador 4 é **aleatório**;
 - Compra a propriedade de maneira aleatória, com probabilidade de 50%.

Caso o jogo demore muito, como é de costume em jogos dessa natureza, o jogo termina na milésima rodada com a vitória do jogador com mais *coins*. Caso mais jogadores terminem com a mesma quantidade de *coins*, o critério de desempate é a ordem de turno dos jogadores nesta partida, sendo a prioridade do último a jogar para o primeiro.

Dados esse padrão de comportamento dos jogadores e das regras de Bankrupt, queremos saber o seguinte:

- Em 300 partidas, qual a distribuição de vitória por comportamento de jogador? Existe algum comportamento que ganha mais que os outros?
- Em média, quantas rodadas o jogo demora para terminar? Quantas partidas de Bankrupt terminam pelo critério de tempo (timeout)?
- Em média quantas rodadas são necessárias para o jogo terminar?

Entrada

Nesta seção definiremos qual o padrão esperado para o arquivo de entrada `gameConfig.txt`, que dita o preço de venda e aluguel das propriedades. Esse arquivo deverá se encontrar na mesma pasta que os outros arquivos fonte do desafio e deverá seguir a risca as instruções a seguir. Seu programa deve ler esse arquivo e configurar as propriedades do jogo baseado nas informações encontradas nele.

Cada linha do arquivo `gameConfig.txt` contém informações a respeito de uma propriedade. Cada linha corresponde a uma propriedade do tabuleiro, descrevendo assim todas as 20 propriedades do jogo.

Cada linha do tabuleiro contém dois valores inteiros. O primeiro é o valor de venda da propriedade e o segundo é o seu valor de aluguel.

Dentro do conteúdo da pasta do desafio encontra-se o arquivo de entrada que deverá ser usado para execução do seu simulador.

Saída

Uma execução do programa proposto deverá rodar 300 simulações de Bankrupt, imprimindo no console os dados referentes às execuções. Esperamos encontrar nos dados as seguintes informações:

- Quantas partidas terminam por time out (1000 rodadas);
- Quantos turnos (rodadas), em média, demora uma partida;
- Qual a porcentagem de vitórias por comportamento dos jogadores;
- Qual o comportamento que mais vence.

Envio da Resposta

Nós da Tapps gostaríamos de analisar sua solução do nosso desafio! Por isso, pedimos para que você crie um **zip** com os arquivos descritos abaixo que compõem sua resposta.

- Código do projeto;
- Um output de saída do projeto (pode ser um arquivo de texto com as informações copiadas da saída da última execução do seu projeto)
- Um arquivo **README** com instruções de execução/compilação, e quaisquer observações que você achar relevante!

O arquivo (.zip) que compõe a resposta do teste devem ser enviados através de um link público (Google Drive, Mega, OneDrive) na resposta deste teste. É extremamente importante que o arquivo seja público para poder ser baixado pelos técnicos da Tapps. Para testar, pode-se tentar baixar o arquivo pelo link em uma aba do navegador no modo anônimo, sem que esteja logado em nenhuma conta.