

**MAC2166 – Introdução à Computação
(Python)
Primeiro Semestre de 2018**

Primeiro Exercício-Programa 1 (EP1)

Data de entrega: **até 23:55 do dia 07 de abril de 2018.**

Previsão do comportamento humano

O estudo da previsibilidade do comportamento humano é um campo intenso de pesquisa, principalmente pelas suas aplicações nas bolsas de valores, em jogos, tratados, nos esportes e para identificar qual aluno desistirá do curso. Recentemente, métodos computacionais vem superando a capacidade humana de prever o comportamento [1].

Neste exercício-programa, implementaremos um programa que “prevê” o comportamento humano utilizando um jogo simples.

O jogo é composto por um jogador humano e o computador. Ambos escolhem ou o valor 0 ou o valor 1. Caso ambos escolham os mesmos números, o computador ganha um ponto. Caso os números escolhidos sejam diferentes, o jogador ganha um ponto. Isso é repetido n vezes. Ganha aquele que acumular mais pontos no final de n jogadas.

A estratégia a ser implementada para que o computador “preveja” o lance do jogador humano é baseada num algoritmo simples de “aprendizagem de máquina”.

Esse aprendizado tem como premissa que cada escolha que o jogador faz é influenciada pela sua jogada anterior. Desta maneira será feita uma contagem: (1) do número de vezes que o jogador humano escolheu 0 dado que no lance anterior seu lance foi 0 (**lance₀₀**), (2) do número de vezes que o jogador humano escolheu 0 dado que no lance anterior seu lance foi 1 (**lance₀₁**), (3) do número de vezes que o jogador humano escolheu 1 dado que no lance anterior seu lance foi 0 (**lance₁₀**), (4) do número de vezes que o jogador humano escolheu 1 dado que no lance anterior seu lance foi 1 (**lance₁₁**).

Então, os seguintes casos podem ocorrer:

Se o lance anterior do jogador foi 0:

1. Se $\text{lance}_{10} > \text{lance}_{00}$: então o computador joga 1
2. Se $\text{lance}_{10} < \text{lance}_{00}$: então o computador joga 0
3. Se $\text{lance}_{10} = \text{lance}_{00}$: então o computador escolhe entre o valor 0 e o valor 1 de forma aleatória

Se o lance anterior do jogador foi 1:

4. Se $\text{lance}_{11} > \text{lance}_{01}$: então o computador joga 1
5. Se $\text{lance}_{11} < \text{lance}_{01}$: então o computador joga 0

6. Se $\text{lance}_{11} = \text{lance}_{01}$: então o computador escolhe entre o valor 0 e o valor 1 de forma aleatória

Obviamente para o primeiro lance do computador, não existe lance anterior do jogador. Então, a escolha do computador é feita de forma aleatória.

Para a escolha aleatória da jogada, faça o seguinte procedimento. Gere um número aleatório utilizando o método das congruências lineares (ver Apêndice). Se o número aleatório for menor ou igual que $2^{32}/2 (=2^{31})$, o computador jogará 0. Caso contrário, jogará 1. Para facilitar a estrutura de seu programa, gere um novo número aleatório no início de cada jogada, e o utilize apenas se necessário conforme as regras acima. Se o número aleatório não for utilizado ele será descartado no início da próxima jogada, quando um novo número for gerado.

Após n jogadas, vence quem obtiver mais pontos (jogador ou computador).

Seu EP deve receber como entrada (via teclado) o número de lances a serem feitos (número inteiro positivo) e a dificuldade do jogo (fácil ou difícil).

No modo “fácil”, o computador não tem nenhum aprendizado, ou seja, todos os lances do computador são aleatórios, utilizando o método das congruências lineares. Se o número aleatório gerado for menor ou igual à $2^{32}/2 (=2^{31})$, o computador lançará 0 e 1, caso contrário.

No modo difícil o computador se valerá do aprendizado descrito anteriormente.

Durante a partida, seu EP deve mostrar graficamente como está o andamento do jogo, isto é, imprimir uma barra que ilustre a pontuação do computador e jogador. A barra de pontuação será simulada pela impressão de asteriscos.

Ex:

JOGADOR: *

MAQUINA: ***

Indica que o jogador venceu um lance e a máquina três.

Ao final do jogo (após os n lances), você deve imprimir na tela quem foi o ganhador ou se houve empate.

Jogue contra o computador e veja se você é capaz de vencê-lo!

Veja abaixo dois exemplos de saída. Seu programa deve fazer a saída exatamente no mesmo formato.

```
Escolha o tipo de jogo (1: Facil; 2: Dificil):2
Entre com o numero de jogadas:5
Faca sua 1a jogada:1
jogador = 1 maquina = 0 Jogador ganha!
JOGADOR: *
MAQUINA:
Faca sua 2a jogada:0
jogador = 0 maquina = 0 Maquina ganha!
JOGADOR: *
MAQUINA: *
Faca sua 3a jogada:1
jogador = 1 maquina = 1 Maquina ganha!
JOGADOR: *
MAQUINA: **
Faca sua 4a jogada:0
jogador = 0 maquina = 0 Maquina ganha!
JOGADOR: *
MAQUINA: ***
Faca sua 5a jogada:0
jogador = 0 maquina = 1 Jogador ganha!
JOGADOR: **
MAQUINA: ***
A maquina venceu!
```

```
Escolha o tipo de jogo (1: Facil; 2: Dificil):1
Entre com o numero de jogadas:5
Faca sua 1a jogada:1
jogador = 1 maquina = 0 Jogador ganha!
JOGADOR: *
MAQUINA:
Faca sua 2a jogada:1
jogador = 1 maquina = 0 Jogador ganha!
JOGADOR: **
MAQUINA:
Faca sua 3a jogada:1
jogador = 1 maquina = 1 Maquina ganha!
JOGADOR: **
MAQUINA: *
Faca sua 4a jogada:1
jogador = 1 maquina = 1 Maquina ganha!
JOGADOR: **
MAQUINA: **
Faca sua 5a jogada:1
jogador = 1 maquina = 0 Jogador ganha!
JOGADOR: ***
MAQUINA: **
Voce venceu!
```

Apêndice:

Método das congruências lineares

Dado um número inicial x_0 , conhecido como semente, o próximo número da sequência é dado por $x_1 = (ax_0 + b) \bmod(m)$. Em geral, o número x_{i+1} é obtido a partir do número x_i pela fórmula $x_{i+1} = (ax_i + b) \bmod(m)$.

Por exemplo, para $a=7$, $b=1$, $m=13$ e $x_0=3$, a sequência de números gerada é: 9, 12, 7, 11, 0, 1, 8, 5, 10, 6, 4, 3, 9, ...

Neste EP, vocês devem usar como parâmetros, os valores $a=22695477$, $b=1$ e $m=2^{32}$, que são utilizados por sistemas conhecidos.

Como imprimir as barras de resultado:

Você pode criar um laço e imprimir o caractere "*". Este laço executa conforme o número de pontos. Para imprimir sem mudar de linha use.

```
print("*", end = "")
```

Observações:

- Todo exercício-programa deve seguir as observações gerais que indicam forma de entrega do exercício e aspectos importantes na avaliação. As observações gerais encontram-se [aqui](#).

Bibliografia

[1] Kanter JM, Veeramachaneni K. Deep feature synthesis: towards automating data science endeavors. IEEE International Conference on Data Science and Advanced Analytics, 2015.