Problema dos Quatro Macacos

Mário Leite

...

O ”Problema dos Quatro Macacos” é bem interessante; em que quatro macacos estão diante de doze bananas com que devem se alimentar, fazendo a distribuição entre eles. Por outro lado, o propósito central é entender como indivíduos ou grupos sociais tomam decisões diante de escassez de recursos e quais estratégias são mais eficazes para garantir a sobrevivência individual ou coletiva. O objetivo desse cenário hipotético é explorar como indivíduos em um grupo lidam com escassez de recursos e quais estratégias eles podem adotar para garantir sua sobrevivência; podendo ser generalizado para um cenário apocalítico em que os indivíduos de uma comunidade devem decidir entre sua humanidade e o instinto de sobrevivência. O problema, aqui relatado levanta várias questões fundamentais:

1. Hierarquia e Poder
   - Como a hierarquia social influencia a distribuição de recursos?
   - Os indivíduos mais fortes ou dominantes têm o direito de monopolizar os recursos?
   - Quais são as consequências de uma hierarquia rígida em tempos de escassez?
2. Comportamento Social
   - Os macacos cooperam ou competem entre si?
   - É possível criar um sistema justo onde todos tenham acesso aos recursos, mesmo que insuficientes?
   - Como surgem conflitos e como eles são resolvidos?
3. Moralidade e Ética
   - Até que ponto é "certo" sacrificar os mais fracos para garantir a sobrevivência dos mais fortes?
   - Qual é o papel da empatia e solidariedade em situações de crise?
   - Esse cenário reflete dilemas humanos em sociedades com recursos limitados?
4. Modelagem Matemática e Ciência da Computação
   - Como podemos modelar esse problema matematicamente? (Ex.: usando “Teoria dos Jogos” ou simulações computacionais)
   - Quais estratégias maximizam a sobrevivência do grupo como um todo?
   - Qual é o equilíbrio ideal entre competição e cooperação?

O programa **“ProblemaQuatroMacacos”**, codificado em Python, é um exemplo de solução para fazer uma simulação da situação de como quatro macacos podem lidar com recurso de apenas uma dúzia de bananas para a alimentação. O ideal seria fazer a divisão normal (12/4) o que daria três bananas para cada um; mas, e se algum deles resolvesse tomar para si os recursos de outro e mesmo apelar para o canibalismo? Isto poderia acontecer com seres humanos em situações extremas de falta de recursos, como no caso de uma catástrofe mundial natural ou mesmo em caso de uma 3ª Guerra Mundial global? Assim, este assunto pode se tornar uma ferramenta poderosa para refletir sobre comportamento social, ética, hierarquia, moralidade e sobrevivência em ambientes com recursos limitados, tal como acontece hoje em algumas regiões do mundo, e pode ser abordado de maneiras diferentes: dependendo do contexto e da realidade que se apresenta (como no caso dos “sobreviventes dos Andes” em outubro de 1972). A **figura 1** mostra o resultado de uma simulação computacional sobre esse assunto, enfocando quatro macacos.

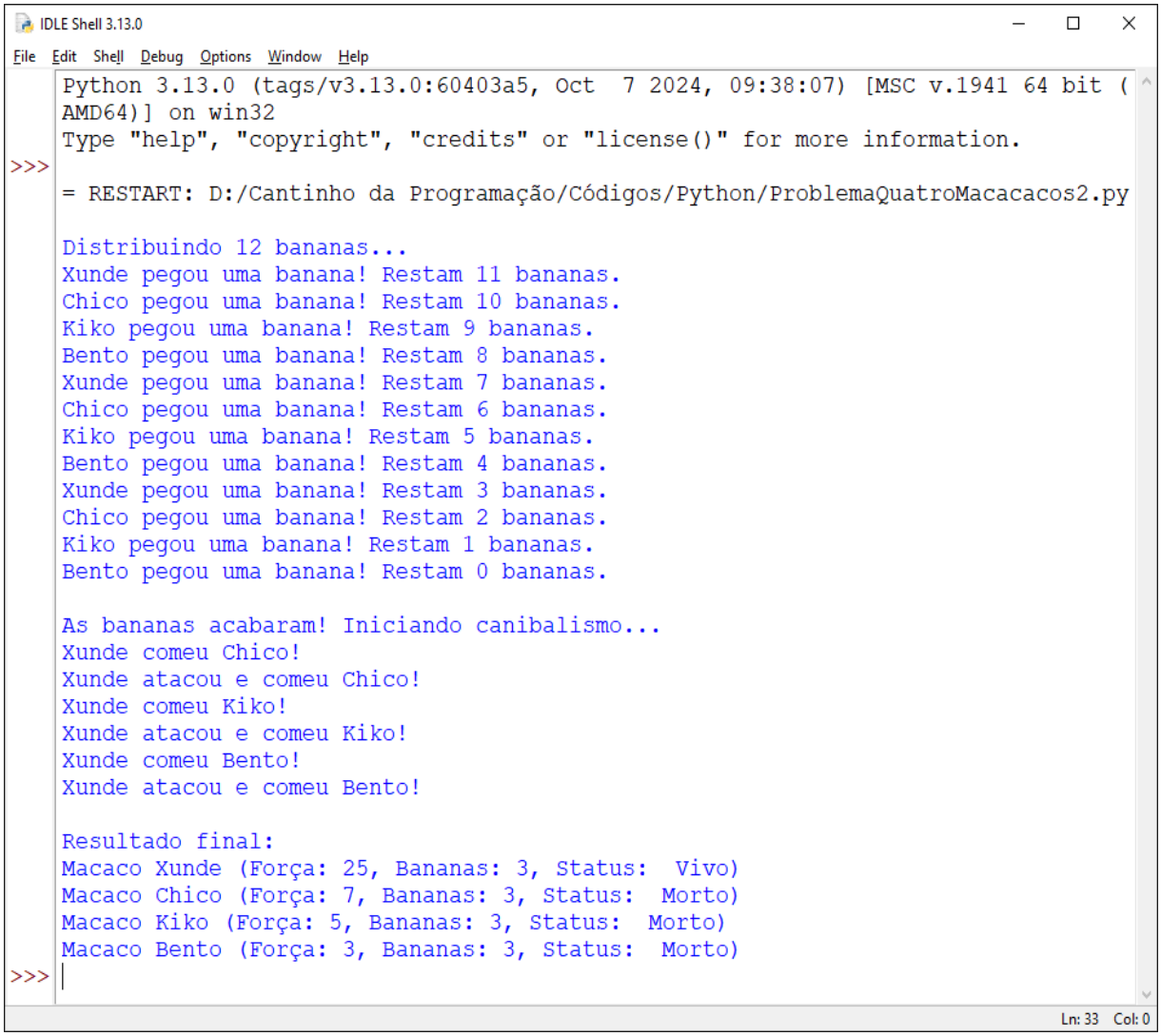

Figura 1 - Uma saída do programa “ProblemaQuatroMacacos”

```
'''
ProblemaQuatroMacacacos2.py
-------------------------------------------------------------------------------
Resolve o clássico "Problema dos Quatro Macacos".
-------------------------------------------------------------------------------
'''
import random

#Classe para representar um macaco
class ClsMacacos:
    def __init__(self, name, força):
        self.name = name  #nome do macaco
        self.força = força  #força do macaco (influencia na hierarquia)
        self.bananas = 0  #número de bananas que o macaco possui
        self.vivo = True #estado do macaco (vivo ou morto)

    def ComerBananas(self):
        if(self.vivo):
            self.bananas += 1

    def ComerMacacos(self, outroMacaco):
        if(self.vivo and not outroMacaco.vivo):
            print(f"{self.name} comeu {outroMacaco.name}!")
            self.força += outroMacaco.força  #ganha força do macaco consumido

    def __str__(self):
        status = " Vivo" if self.vivo else " Morto"
        return f"Macaco {self.name} (Força: {self.força}, Bananas:
               {self.bananas}, Status: {status})"
#-------------------------------------------------------------------------------
#Função para distribuir bananas
def DistribuirBananas(LstMacacos, bananas):
    print(f"Distribuindo {bananas} bananas...")
    while(bananas > 0):
        #Ordena os macacos pela força (hierarquia)
        LstMacacos.sort(key=lambda macaco: macaco.força, reverse=True)

        for macaco in LstMacacos:
            if(bananas > 0 and macaco.vivo):
                macaco.ComerBananas()
                bananas -= 1
               print(f"{macaco.name} pegou uma banana! Restam {bananas} bananas.")
            else:
                break
#-------------------------------------------------------------------------------
#Função para simular canibalismo
def SimularCanibalismo(LstMacacos):
    print("\nAs bananas acabaram! Iniciando canibalismo...")
    LstMacacos.sort(key=lambda macaco: macaco.força, reverse=True)
    #ordena pelo mais forte

    for i, macaco in enumerate(LstMacacos):
        if(macaco.vivo):
            for j in range(i+1, len(LstMacacos)):
                outroMacaco = LstMacacos[j]
                if(outroMacaco.vivo and macaco.força > outroMacaco.força):
                    outroMacaco.vivo = False
                    macaco.ComerMacacos(outroMacaco)
                    print(f"{macaco.name} atacou e comeu {outroMacaco.name}!")
#-------------------------------------------------------------------------------
```

```python
#Função principal
def main():
    #Criação da lista dos macacos
    LstMacacos = [
        ClsMacacos("Xunde", 10), #macaco dominante
        ClsMacacos("Chico", 7),  #macaco subordinado
        ClsMacacos("Kiko", 5),   #macaco fraco
        ClsMacacos("Bento", 3)   #macaco mais fraco
    ]

    print()
    #Número total de bananas
    TotalBananas = 12

    #Distribui as bananas
    DistribuirBananas(LstMacacos, TotalBananas)

    #Simula canibalismo
    SimularCanibalismo(LstMacacos)

    #Mostra o resultado final
    print("\nResultado final:")
    for macaco in LstMacacos:
        print(macaco)
#=============================================================================
#Programa principal)
if(__name__ == "__main__":
    main()
#Fim do programa "ProblemaQuatroMacacacos2" ----------------------------------
```