



AMAZONAS
GOVERNO DO ESTADO



UNIVERSIDADE DO ESTADO DO AMAZONAS- UEA

ESCOLA DE TECNOLOGIA – EST

ATIVIDADE DA DISCIPLINA LABORATÓRIO DE PROGRAMAÇÃO DE
COMPUTADORES I

DAWKIN'S WEASEL PROGRAM

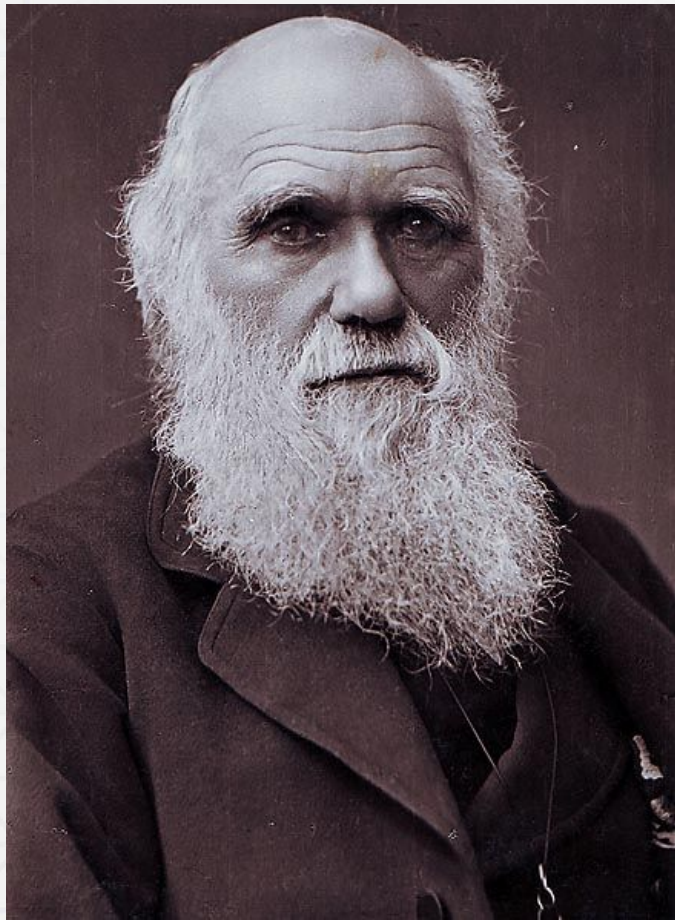
DISCENTES:

DÁRIO SOUZA ROCHA FILHO

JOSÉ CLEMENTE SIMÕES DE QUEIROZ

LEONARDO CASTRO DA SILVA

Teoria Da Evolução



Charles Darwin(1809-1882)

- Filme Criação(2009)
- Religiosidade
- Enfermidade
- Perdas

Teoria Da Evolução

- Ancestrais em comum.
- Indivíduos da mesma espécie, mesmo que parentes próximos, possuem variações entre si
- Variações hereditárias.
- Recursos escassos geram luta pela sobrevivência.
- Variações vantajosas permitem que certos indivíduos se destaquem e outros não.

Teoria Da Evolução

- Sobrevivência dos mais aptos.
- Os mais aptos podem transmitir às próximas gerações a característica vantajosa, caso seja hereditária.
- Seleção Natural
- Seleção Artificial, gerada pela influência dos humanos em determinados organismos.

Dawkin's Weasel Program: Funcionamento

```
Weasel.py X
Weasel.py > ...
1 # pegando a biblioteca random
2 import random
3
4 # INICIANDO AS VARIÁVEIS QUE SERÃO UTILIZADAS
5 frase_alvo = "METHINKS IT IS LIKE A WEASEL"
6 frase_mais_parecida = ''
7 score = 0
8 maior_score = 0
9 frase_pai = []
10 geracao = 0
11
12 # GERANDO A PRIMEIRA FRASE QUE DARÁ ORIGEM À PRIMEIRA GERAÇÃO DE 100 FILHAS
13 for i in range(28):
14     frase_pai.append(random.choice("ABCDEFGHIIJKLMNOPQRSTUVWXYZ "))
15
16 # LOOP RESPONSÁVEL POR VERIFICAR A NECESSIDADE DE CONTINUAR A GERAÇÃO
17 # DE FRASES FILHAS
18 while maior_score < 28:
19     geracao += 1
20
21     # LOOP RESPONSÁVEL POR GERAR AS FRASES FILHAS
22     for filha in range(100):
23         frase_filha = ''
24
25         # LOOP RESPONSÁVEL POR GERAR UMA ÚNICA FILHA
26         # E FAZER A CONTAGEM DE PONTUAÇÃO(SCORE)
27         for j in range(28):
28             numero_aleatorio = random.randint(1, 100)
29
30             if numero_aleatorio <= 5:
```

```
36         frase_filha += random.choice(
37             "ABCDEFGHIIJKLMNOPQRSTUVWXYZ ")
38
39         else:
40
41             frase_filha += frase_pai[j]
42
43         if frase_filha[j] == frase_alvo[j]:
44
45             score += 1
46
47         if maior_score < score:
48
49             frase_mais_parecida = frase_filha
50             maior_score = score
51
52         if maior_score == 28:
53
54             break
55
56         print(frase_filha, end="; ")
57         print("Geração:", end=" ")
58         print(geracao, end="; ")
59         print("Filha: " + str(filha), end="; ")
60         print("Score: " + str(score))
61         score = 0
62
63     frase_pai = frase_mais_parecida
64
65 # EXPOSIÇÃO DO RESULTADO OBTIDO
66 print(frase_mais_parecida, end="; ")
67 print("Geração:", end=" ")
68 print(geracao, end="; ")
69 print("Filha: " + str(filha), end="; ")
70 print("Score: " + str(score))
71
```

Dawkin's Weasel Program: Funcionamento

Weasel.py X

Weasel.py > ...

```
1  # pegando a biblioteca random
2  import random
3
4  # INICIANDO AS VARIÁVEIS QUE SERÃO UTILIZADAS
5  frase_alvo = "METHINKS IT IS LIKE A WEASEL"
6  frase_mais_parecida = ''
7  score = 0
8  maior_score = 0
9  frase_pai = []
10 geracao = 0
11
12 # GERANDO A PRIMEIRA FRASE QUE DARÁ ORIGEM À PRIMEIRA GERAÇÃO DE 100 FILHAS
13 for i in range(28):
14     frase_pai.append(random.choice("ABCDEFGHIJKLMNOPQRSTUVWXYZ "))
15
16
```

Dawkin's Weasel Program: Funcionamento

```
28 # LOOP RESPONSÁVEL POR GERAR UMA ÚNICA FILHA
29 # E FAZER A CONTAGEM DE PONTUAÇÃO(SCORE)
30 for j in range(28):
31     numero_aleatorio = random.randint(1, 100)
32
33     if numero_aleatorio <= 5:
34
35         frase_filha += random.choice(
36             "ABCDEFGHIJKLMNOPQRSTUVWXYZ ")
37
38     else:
39
40         frase_filha += frase_pai[j]
41
42     if frase_filha[j] == frase_alvo[j]:
43
44         score += 1
45
46
```

Dawkin's Weasel Program: Funcionamento

```
17 # LOOP RESPONSÁVEL POR VERIFICAR A NECESSIDADE DE CONTINUAR A GERAÇÃO
18 # DE FRASES FILHAS
19 while maior_score < 28:
20
21     geracao += 1
22
23     # LOOP RESPONSÁVEL POR GERAR AS FRASES FILHAS
24     for filha in range(100):
25
26         frase_filha = ''
27
28         # LOOP RESPONSÁVEL POR GERAR AS FRASES FILHAS
29         for i in range(28):
30
31             # GERAÇÃO DE UMA FRASE FILHA
32             frase_filha += random.choice('ABCDEFGHIJKLMNOPQRSTUVWXYZ')
33
34         # VERIFICAÇÃO DA FRASE FILHA
35         score = 0
36         for i in range(28):
37
38             if frase_filha[i] == frase_pai[i]:
39                 score += 1
40
41         # ATUALIZAÇÃO DA FRASE PAI
42         frase_pai = frase_filha
43
44         # ATUALIZAÇÃO DO MAIOR SCORE
45         if maior_score < score:
46
47             frase_mais_parecida = frase_filha
48             maior_score = score
49
50         # VERIFICAÇÃO DO MAIOR SCORE
51         if maior_score == 28:
52
53             break
54
55     # IMPRIMIR O RESULTADO DA GERAÇÃO
56     print(frase_filha, end="; ")
57     print("Geração:", end=" ")
58     print(geracao, end="; ")
59     print("Filha: " + str(filha), end="; ")
60     print("Score: " + str(score))
61     score = 0
62
63     frase_pai = frase_mais_parecida
64
65 # EXPOSIÇÃO DO RESULTADO OBTIDO
66 print(frase_mais_parecida, end="; ")
67 print("Geração:", end=" ")
68 print(geracao, end="; ")
69 print("Filha: " + str(filha), end="; ")
70 print("Score: " + str(score))
```


Dawkin's Weasel Program: Funcionamento

```
EETHINKS IT IS LIKE A WEASDL; Gera: 94; Filha: 82; Score: 26
EETHITKS IT IS LLKV A WEASML; Gera: 94; Filha: 83; Score: 23
EZTHINKS IT IS LIKE A WEASEL; Gera: 94; Filha: 84; Score: 26
EETHINKS IT IS LIKS A WEASEL; Gera: 94; Filha: 85; Score: 26
EETHINKS IT IS FIKE A WETSEL; Gera: 94; Filha: 86; Score: 25
EETHINKS IT IS LIKE A WETSEL; Gera: 94; Filha: 87; Score: 26
ONTHINKS IT IS LIKE A WEASEL; Gera: 94; Filha: 88; Score: 26
EETHINKS IT ISCLIKE A WEASEL; Gera: 94; Filha: 89; Score: 26
EETHINKS IT IS LIKE A WEASEL; Gera: 94; Filha: 90; Score: 27
EETHINKS IT IS LIKE A WEASEL; Gera: 94; Filha: 91; Score: 27
EETHINKS IT IS LIKE A WEASEQ; Gera: 94; Filha: 92; Score: 26
EEDHINKS IT IS LIKE A WEAQEL; Gera: 94; Filha: 93; Score: 25
EETHINKS ST IS LIKE A WEASEL; Gera: 94; Filha: 94; Score: 25
EETHINKS IT ISELIKE A WEISEL; Gera: 94; Filha: 95; Score: 25
EETHINKS IT IS LIKE AZWEASEL; Gera: 94; Filha: 96; Score: 26
EETHINKS IT IS LIKE APWEASEL; Gera: 94; Filha: 97; Score: 26
AETHINKSTIT IS LLKE A WEASEL; Gera: 94; Filha: 98; Score: 25
EETHINKS IT IS LIKE A WEASEL; Gera: 94; Filha: 99; Score: 27
EETHINKS IT IM LIKE U WEASEL; Gera: 95; Filha: 0; Score: 25
EQTHINKS IT IS LIZE A WEASEL; Gera: 95; Filha: 1; Score: 25
EETHINKSOIT IS LIZE A WEASEM; Gera: 95; Filha: 2; Score: 24
EETHINKS IT IS LIKC A WEASEL; Gera: 95; Filha: 3; Score: 26
EETHINKS IT IS LIQE A WEASEL; Gera: 95; Filha: 4; Score: 26
EETZINKS IT IS LIKE A WEASEL; Gera: 95; Filha: 5; Score: 26
EETHINKS IT IS LIKE A WEASEL; Gera: 95; Filha: 6; Score: 27
EETHINKS IT IS LIKE K WEPSEL; Gera: 95; Filha: 7; Score: 25
EETHINKS IT ISFLIKEYA WEASEL; Gera: 95; Filha: 8; Score: 25
EETHINKS IT IS LIKE A WEASEL; Gera: 95; Filha: 9; Score: 27
EETHINKS IT IS LIKE A WEASEL; Gera: 95; Filha: 10; Score: 27
EETHINKS IT PS LIKE A WEASEL; Gera: 95; Filha: 11; Score: 26
METHINKS IT IS LIKE A WEASEL; Gera: 95; Filha: 12; Score: 28
```

[Done] exited with code=0 in 1.036 seconds

Relação Programa - Teoria

- Biodiversidade/competição.
- 5% chance de desenvolver uma mutação.
- Realidade na natureza.
- Weasel program.
- Trapaça no código em relação a natureza.

Relação Programa - Teoria



- Bicho-Pau



- Bicho-Folha

OBRIGADO!