
PPPD - Lab. 05

Copyright ©2021 M. Śleszyńska-Nowak i in.

Zadanie punktowane, lab 05, grupa A, 2021/2022, autor: Michał Denkwicz

Uwaga: w rozwiązaniu zadania nie można używać list.

Temat: Symulacja stad wilków.

Treść zadania

Celem jest napisanie programu do symulowania dwóch stad wilków rywalizujących o terytoria łowieckie, gdzie polują na zające.

Zasady symulacji

Symulacja polega na śledzeniu, dzień po dniu zachowania dwóch stad wilków polujących na zające na częściowo współdzielonym terytorium. Przebieg pojedynczej symulacji jest następujący: codziennie, przez **n_dni** dni wilki z obu stad wychodzą na polowanie. Stado może też wysłać jednego lub więcej wilków (są to szczególnie silne i agresywne osobniki) do “ataku” (zaburzenia polowania drugiego stada i podkradnięcia mu zdobyczy). Na początku dnia każde stado, z prawdopodobieństwem odp. **p1** dla pierwszego stada i **p2** dla drugiego, wysyła jednego wilka do ataku. Dodatkowo jeżeli przez 2 poprzednie dni z rzędu stado nie wysyłało żadnego wilka do ataku, stado wysyła dodatkowego wilka. Następnie obliczany jest rezultat polowania każdego stada. Symulacja śledzi wyniki polowań i na końcu wylicza średnią dzienną dla obu stad.

Polowanie rozstrzyga się w sposób następujący:

- Jeżeli własne stado wysłało więcej wilków do ataku, niż stado rywali, polowanie przynosi losowo od 30 do 50 zające.
- Jeżeli to rywale wysłali więcej wilków do ataku, polowanie przynosi od 5 do 20 zające.
- Jeżeli oba stada wysłały tyle samo wilków do ataku, dochodzi do długiej i chaotycznej walki; stado udaje się upolować tylko 5 zające.
- Jeżeli żadne stado nie wysłało atakujących, liczbę zdobyczy określa parametr **k** (będący parametrem symulacji).

Działanie programu

Na początku programu użytkownik powinien być poproszony o podanie trybu działania (wybór przez wprowadzenie cyfry 1, 2 lub 3):

1. *Polowanie:* Użytkownik jest poproszony o podanie **p1**, **p2**, **k** oraz **n_dni**. Następnie wykonuje się symulacja polowania, której wynik (średnia dzienna liczby upolowanych zające dla każdego ze stad) zostaje wypisany na ekran.
2. *Tabela:* Użytkownik podaje parametry **r** oraz **k**, są to liczby całkowite dodatnie. Program wykonuje funkcję **tabela** (opis poniżej).
3. Kończy działania programu.

Po każdej operacji menu powinno wyświetlić się ponownie, do momentu aż użytkownik wyda polecenie zakończenia. Podanie błędnych danych kończy natychmiast działanie programu.

Funkcja `tabela(r, k)`: Dla liczb całkowitych t i r możemy wyznaczyć prawdopodobieństwo ataku (tzn. p_1 czy p_2), wg wzoru: $\frac{1}{1+\exp(-\frac{5t}{2r+1})}$. Funkcja dla każdej pary liczb całkowitych t_1 i t_2 z przedziału $[-r, r]$ przeprowadza symulację 100 dni polowań. Parametrami symulacji są podane k , oraz p_1 i p_2 obliczone dla danych t_1 i t_2 . W trakcie działania funkcja zapisuje do pliku `wynik.txt` tabelę zawierającą wyniki (pary średnich) dla danych poziomów agresji (przykład wynikowego pliku znajduje się na Moodle):

- Każdy wiersz tabeli odpowiada jednej wartości t_1 , a kolumna - wartości drugiego t_2 .
- Każdy wiersz powinien zaczynać się od prawdopodobieństwa ataku (p_1) dla danego wiersza.

Na końcu funkcja wypisuje na ekran wartości p_1 i p_2 dla których suma średnich zdobyczy była największa, wraz z tą sumą.

Stworzyć należy (przynajmniej) 4 funkcje:

1. `polowanie(n_swoich, n_rywali, k)` – zwraca liczbę upolowanych zajęcy, zgodnie z opisem powyżej.
2. `symulacja(p1, p2, n_dni, k)` – zwraca 2 wartości: średnie (dienne) zajęcy upolowanych przez jedno i drugie stado.
3. `p_ataku(t, r)` – zwraca prawdopodobieństwo ataku zgodnie z opisem powyżej.
4. `tabela(r, k)` – Przeprowadza symulacje w.g. opisu powyżej.

Uwaga: Funkcje te powinny mieć dokładnie takie nazwy, parametry i zwracane wartości.

Ziarno generatora liczb losowych należy na początku programu ustawić na 202121. Przykłady interakcji użytkownika z programem znajdują się na końcu opisu zadania. Przykładowy plik wynikowy znajduje się na Moodle.

Podpowiedź: Aby wypisać liczbę zmiennoprzecinkową zadaną precyzją, np. zmienną x z precyzją 3 cyfry po przecinku, należy użyć następującego wyrażenia:

```
x = 12.34545635
print(f'{x:.3f}')
>>> 12.345
```

Aby wypisać liczbę całkowitą w polu o zadanej szerokości (np. 5), należy użyć następującego wyrażenia:

```
k = 123
print(f'aaa{k: 5d}aaa')
>>> aaa 123aaa
```

Estetyczne kolumny przy wypisywaniu można uzyskać wstawiając znaki tabulacji:

```
print(f'{12.51}\t{1.41}\t{56.11}\t{0.01}')
print(f'{0.51}\t{19.41}\t{5.11}\t{10.01}')
# Spowoduje wypisanie:
>>> 12.51    1.41    56.11    0.01
>>> 0.51    19.41    5.11    10.01
```

Punktacja

Za poszczególne elementy można uzyskać następującą liczbę punktów:

- Obsługa menu: 1p
- Funkcjonalność pojedynczej symulacji: 4p
- Funkcje `p_ataku` i `polowanie`: 2p
- Funkcja `tabela` (w tym zapis do pliku): 3p

Uwaga

- Jeśli rozwiązanie nie spełnia postawionych wymagań (korzysta z list albo nie zawiera opisanych funkcji), zadanie jest oceniane na 0 punktów.

- Jeśli program się nie kompiluje (interpretuje), ocena jest zmniejszana o połowę.
- Jeśli kod programu jest niskiej jakości (nieestetycznie formatowanie, mylące nazwy zmiennych itp.), ocena jest zmniejszana o 2p.

Przykłady interakcji użytkownika z programem

```
Co chcesz zrobić? (1 - Symulacja, 2 - Tabela, 3 - Koniec): 1
Podaj p1: 0.9
Podaj p2: 0.15
Podaj k: 10
Podaj liczbę dni: 200
Stado 1 upolowało 28.9 zajęcy/dzień, a stado 2 28.7 zajęcy/dzień.
Co chcesz zrobić? (1 - Symulacja, 2 - Tabela, 3 - Koniec): 2
Podaj k: 30
Podaj r: 2
Najlepsza suma 0.881, 0.119 (59.580)
Co chcesz zrobić? (1 - Symulacja, 2 - Tabela, 3 - Koniec): 3
```

Process finished with exit code 0

Plik wynikowy:

```
p1=0.119: 26.1, 25.5 26.4, 26.3 22.1, 22.5 17.7, 17.2 15.0, 14.2
p1=0.269: 26.7, 26.6 26.1, 25.9 22.4, 21.9 15.5, 15.7 13.6, 13.4
p1=0.500: 27.4, 26.7 24.9, 26.1 22.5, 22.4 16.2, 16.0 13.1, 12.8
p1=0.731: 28.8, 29.7 27.9, 27.3 22.6, 22.0 14.8, 15.9 9.8, 10.0
p1=0.881: 29.8, 29.8 26.2, 26.1 20.3, 20.1 15.3, 15.1 10.8, 10.6
```

Niepoprawne dane:

```
Co chcesz zrobić? (1 - Symulacja, 2 - Tabela, 3 - Koniec): 1
Podaj p1: -1
Podaj p2: 3.3
Podaj k: 10
Podaj liczbę dni: 100
```

Traceback (most recent call last):

```
File "/home/michade/proj/PPPD/PPPD_21_22/LAB zadania punktowane/Lab05/MAT/B/2021-MAT-05B.py", line 14
    main()
```

```
File "/home/michade/proj/PPPD/PPPD_21_22/LAB zadania punktowane/Lab05/MAT/B/2021-MAT-05B.py", line 13
    s1, s2 = symulacja(p1, p2, n_dni, k)
```

```
File "/home/michade/proj/PPPD/PPPD_21_22/LAB zadania punktowane/Lab05/MAT/B/2021-MAT-05B.py", line 81
    raise ValueError('Niepoprawna wartość')
```

ValueError: Niepoprawna wartość

Process finished with exit code 1