

Zadanie Programistyczne 2: Cykle

Maksymilian Neumann

June 21, 2022

1 Implementacja Fisher-Yates shuffles

Funkcja przyjmuje listę w postaci $[1, 2, \dots, n]$ (element neutralny) i zwraca jej losową permutację.

```
1 import random
2
3 def shuffle(given):
4     permutation = []
5     for i in range(len(given)):
6         temp = random.randint(0, (len(given)-1))
7         permutation.insert(0, given[temp])
8         given.pop(temp)
9     return permutation
```

Listing 1: Implementacja Fisher-Yates shuffles

2 Rozkład Permutacji na Cykle

Funkcja przyjmuje permutacje w postaci listy i zwraca rozkład jej na cykle w postaci listy list.

```
1 def cycler(permuation):
2     cycles = []
3     for i in range(1, len(permuation) + 1):
4         if permuation[i - 1] != 0:
5             temp = i
6             cycle = []
7             cycle.append(i)
8             while permuation[temp - 1] != i:
9                 cycle.append(permuation[temp - 1])
10                temp = permuation[temp - 1]
11            cycles.append(cycle)
12            for j in cycle:
13                permuation[permuation.index(j)] = 0
14    return cycles
```

Listing 2: Rozkład Permutacji na cykle

3 Test Działania Kodu

Z wykładu dowiedzieliśmy się ,że dla $\pi \in S_n$,gdzie $L(\pi)$ = liczba cykli permutacji π

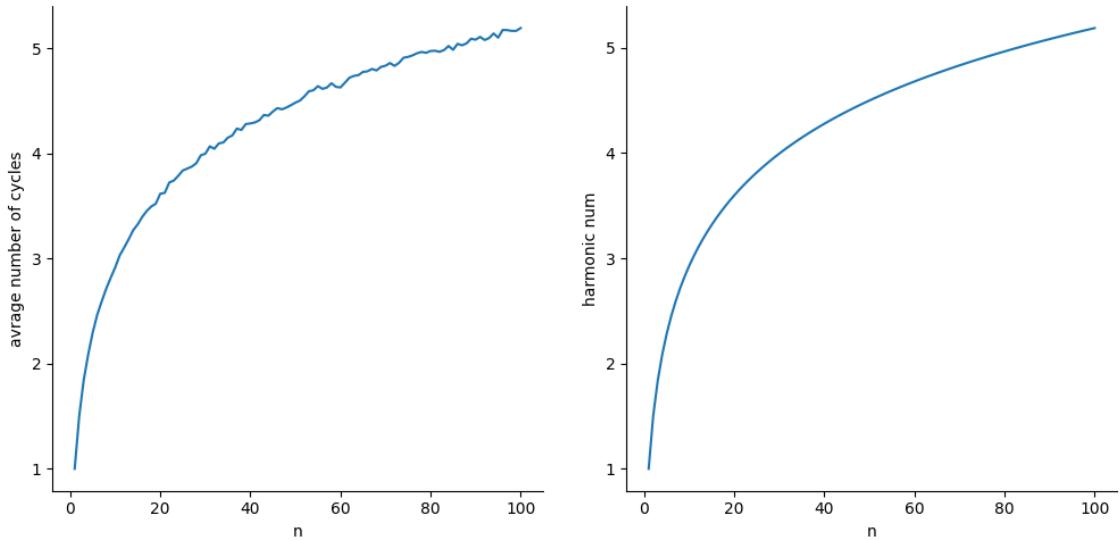
$$E(L) = H_n$$

która będzie naszą oczekiwanaą wartością

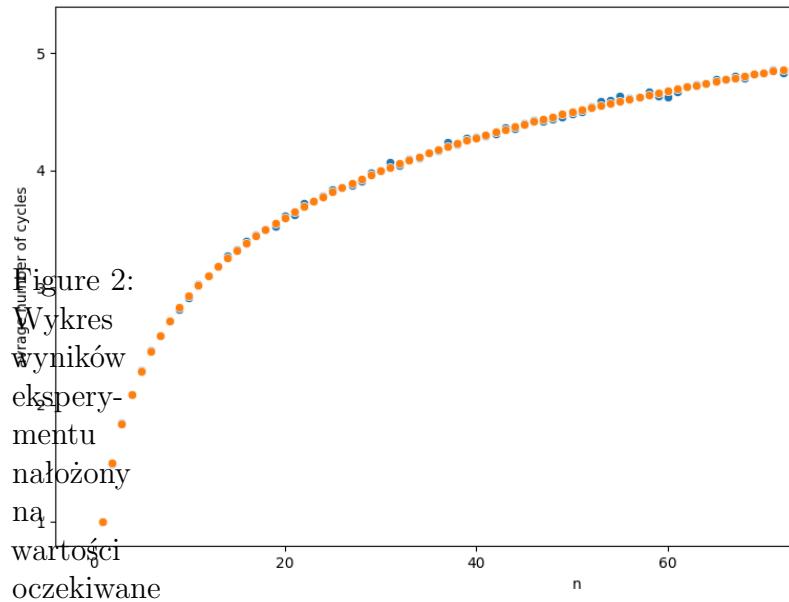
```
1 from statistics import mean
2
3 def test():
4     f = open("test.csv", 'w')
5     f.write("n;avrage number of cycles cycles\n")
6     for n in range(1, 101):
7         lengths = []
8         for i in range(8000):
9             x = list(range(1, n + 1))
10            lengths.append(len(cycler(shuffle(x))))
11            row = str(n) + ";" + str(mean(lengths)) + "\n"
12            f.write(row)
13    f.close()
```

Listing 3: Test zapisyjący wyniki do pliku

Figure 1: Lewy wykres przedstawia wyniki eksperymentu, Prawy wartości oczekiwane



ht



Wyniki eksperymentu sugerują poprawność kodu.

4 Główne Zadanie

Hipoteza:

Można zauważyc, że dla $\pi \in S_n$, gdzie $\pi = C_1 \circ \dots \circ C_k$

$$1. M_n = \max(\text{len}(C_1), \dots, \text{len}(C_k)) = \lceil \frac{n}{k} \rceil$$

2. Wiemy również że takich maksymalnych długości jest $\binom{n}{k}$

Z 1. i 2. wynika:

$$E(M_n) = \frac{\sum_{k=1}^n \lceil \frac{n}{k} \rceil \binom{n}{k}}{n!}$$

Eksperiment:

```

1 from statistics import mean
2
3 def zadanie():
4     f = open("zadanie.csv", 'w')

```

```
5     f.write("n;avrage max length of a cycle in a permutation\\
6     n")
7     for n in range(1, 101):
8         maxLengths = []
9         for i in range(8000):
10            x = list(range(1, n + 1))
11            maxLengths.append(max(len(c) for c in cycler(
12                shuffle(x))))
13            row = str(n) + ";" + str(mean(maxLengths)) + "\n"
14            f.write(row)
15        f.close()
```

Listing 4: Eksperiment numeryczny dla głównego zadania