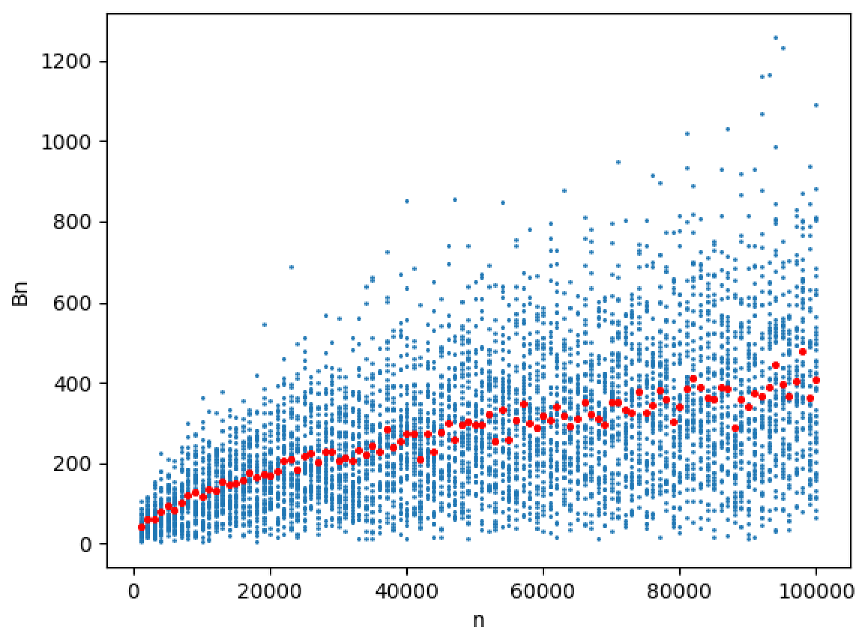
# Metody Probabilistyczne i Statystyka

# Homework2

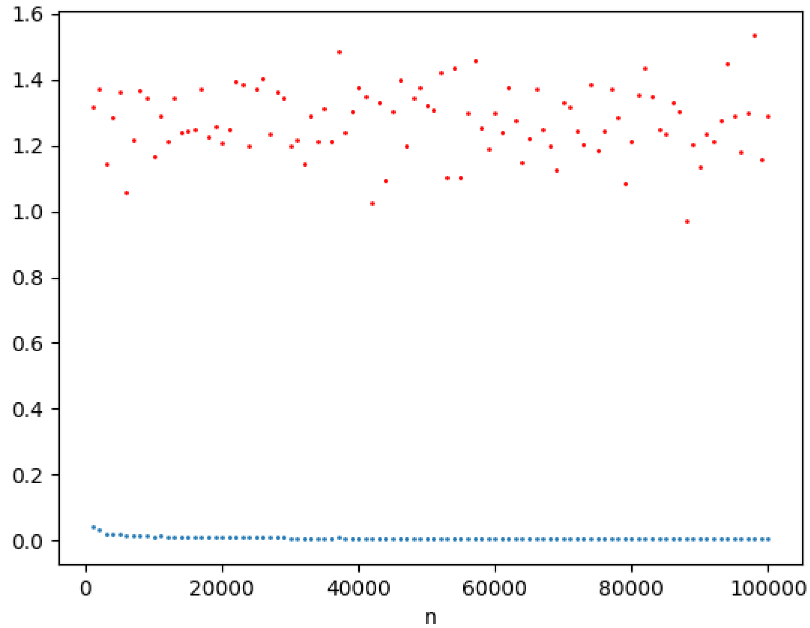

# Krystian Musiałczyk 268437


Do zrealizowania zadania wykorzystałem język Python. Kod źródłowy znajduje się w pliku code.txt.

a) Bn – moment pierwszej kolizji. Błąd aproksymacji rośnie wraz ze wzrostem liczby losowych punktów. Znacznie lepszą aproksymację wartości średniej uzyskujemy na drugim wykresie, przedstawiającym b(n)/n.




Ilorazy b(n)/n - punkty niebieskie oraz b(n)/sqrt(n) - punkty czerwone




Wykres b(n)/n jest bardzo zbliżony do funkji stałej, a to oznacza, że b(n) jest w przybliżeniu funkcją liniową n.

Pomimo bardzo dużej ilości urn, momenty pierwszych kolizji nie mają dużych wartości. Ten podpunkt zawiera w sobie paradoks dnia urodzin. Problem wziął swoją nazwę od następującego pytania:

Ile minimalnie osób należy wybrać, żeby prawdopodobieństwo znalezienia wśród nich co najmniej dwóch osób obchodzących urodziny tego samego dnia było większe od 0,5 ?

Odpowiedź brzmi 23.

Ta zaskakująco mała liczba osób jest przyczyną określenia „Paradoks dnia urodzin".

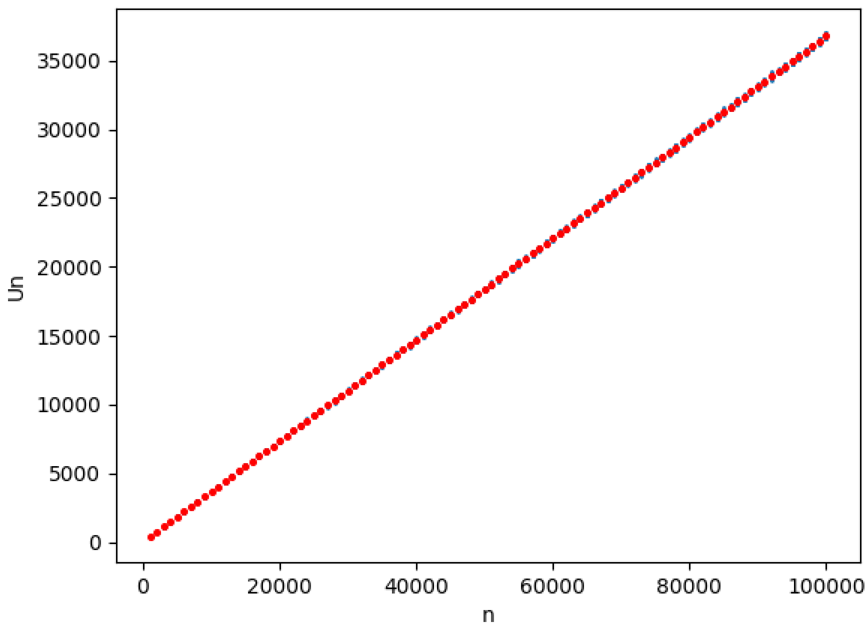
Znaczenie birthday paradox w kontekście funkcji haszującej I kryptograficznej funkcji haszującej:

Z paradoksu urodzinowego korzysta tak zwany atak urodzinowy. Celem ataku urodzinowego jest znalezienie kolizji funkcji haszującej. Atak korzysta z metody brute force, szukając dwóch haseł przyporządkowanych temu samemu indeksowi. Biorąc pod uwage birthday paradox jest to zadanie dużo prostrze niż wskazuje na to intuicja. Znalezienie dwóch haseł o tym samym indeksie pozwala odkowodwać algorytm generujący kolizję.

Źródło: wikipedia.org

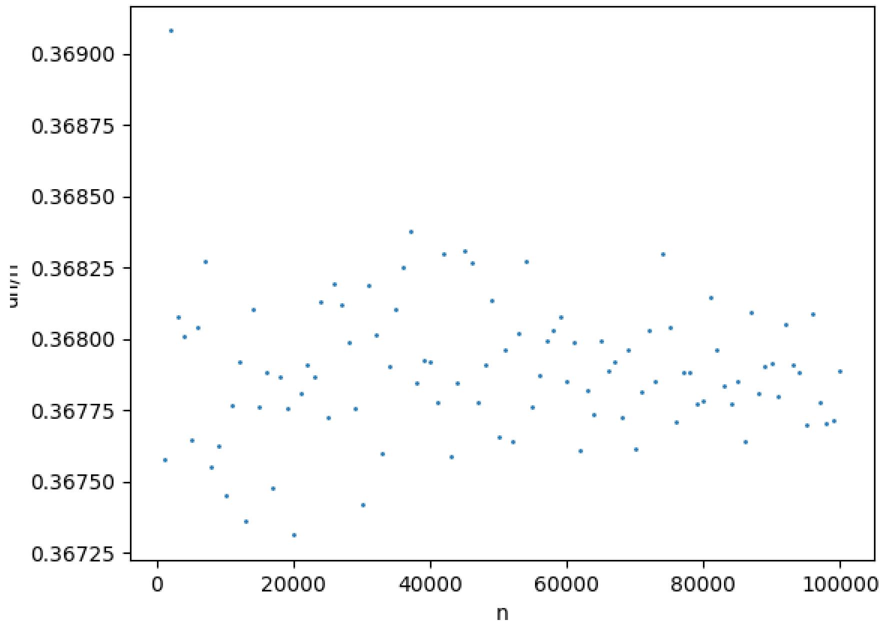
b) Un – liczba pustych urn po wrzuceniu n kul. Un rośnie wraz ze wzrostem liczby losowych punktów.

Un jest funkcją liniową n.

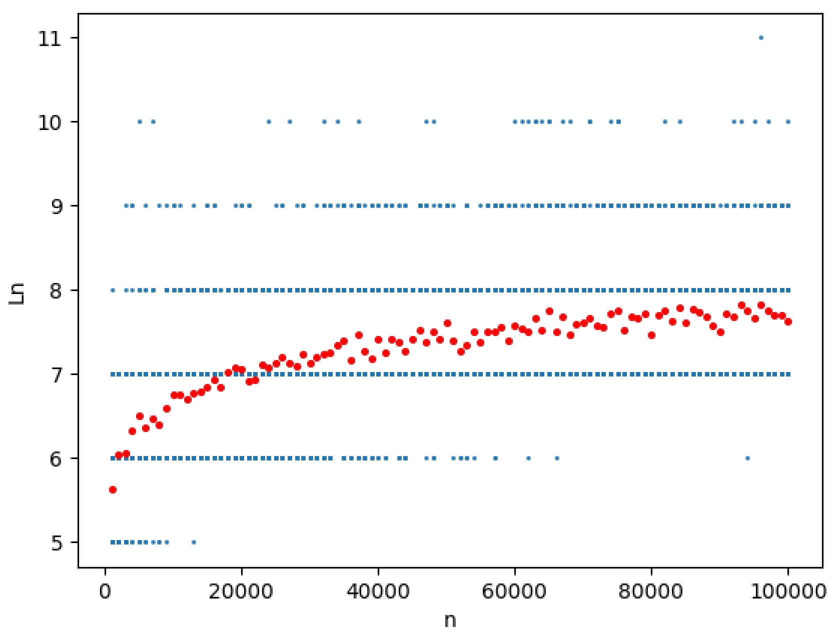



Wykres u(n)/n.  Skala wartości ma tutaj bardzo dużą precyzję. Prawie wszystkie punkty mieszczą się w przedziale (0.36825 , 0,36750). Zaokrąglam wynik do 0.368. Z równania u(n)/n = 0.368 otrzymujemy przybliżoną wartość:

Un = 0.368 * n

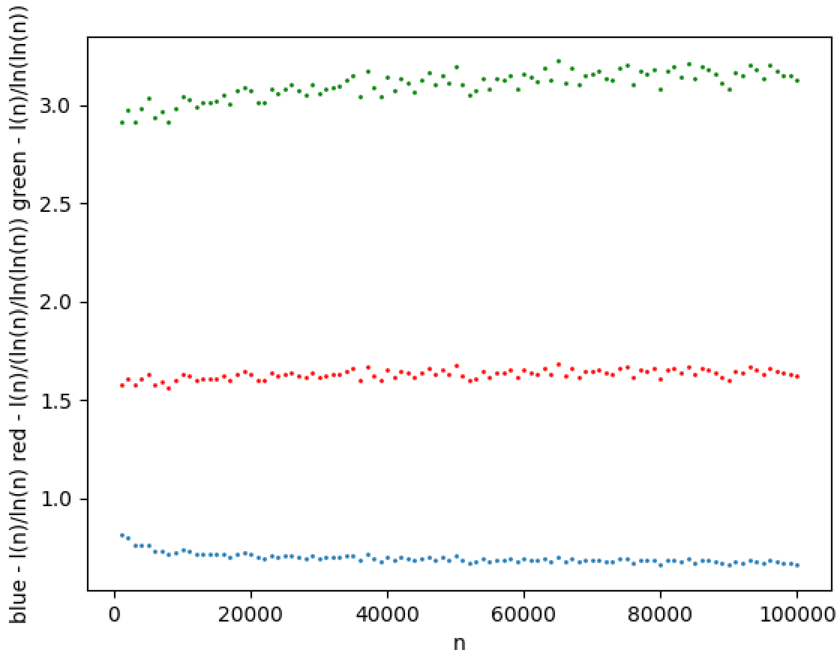



c) Ln - maksymalna liczba kul w urnie po wrzuceniu n kul. Koncentracja punktów wokół wartości średniej jest słaba.

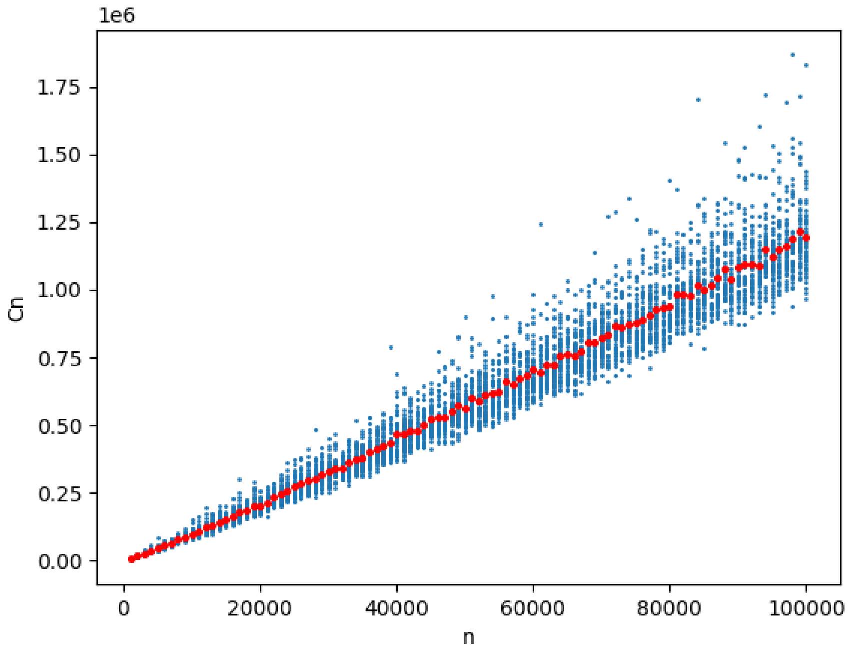



Wykres l(n)/ln(n) - punkty niebieskie, l(n)/ln(n)/ln(ln(n)) - punkty czerwone oraz l(n)/ln(ln(n)) - punkty zielone.

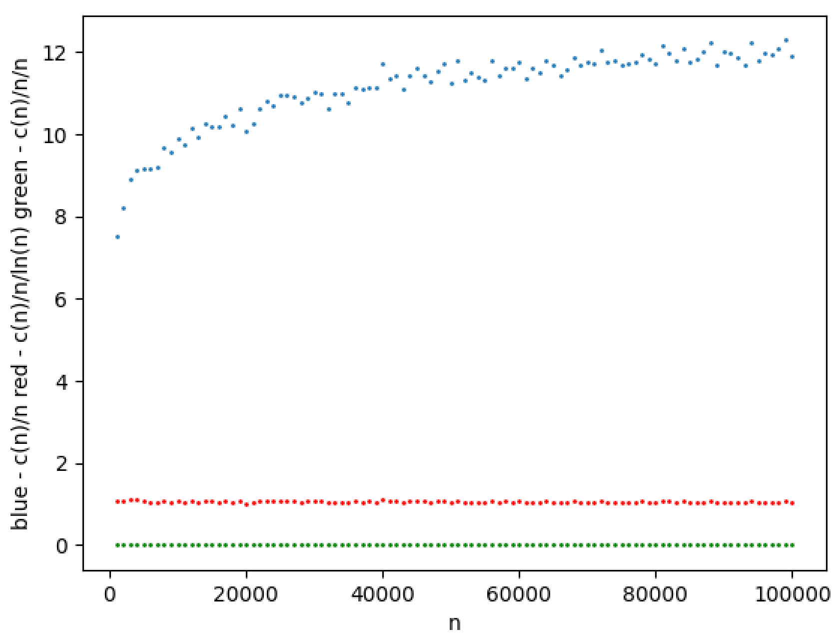
Punkty czerwone najbardziej przypominają funkcję stałą,  co pozwala mi stwierdzić zależność:
Ln = O(ln(n)*ln(ln(n)))



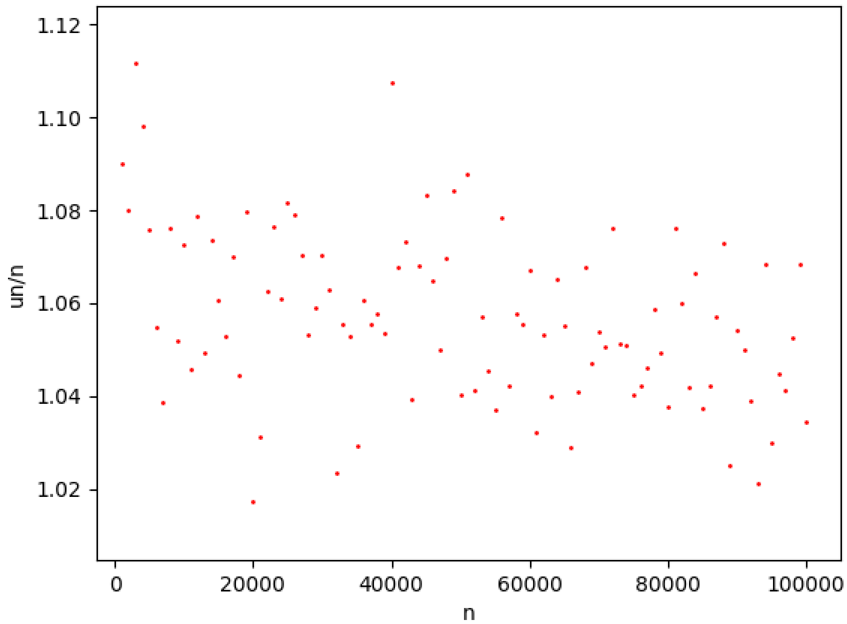

d) Cn – minimalna liczba rzutów, po której w każdej z urn jest co najmniej jedna kula. Koncentracja punktów wokół wartości średniej słabnie wraz ze wzrostem liczby losowych punktów n. Wartości średnie sugerują funkcję liniową.

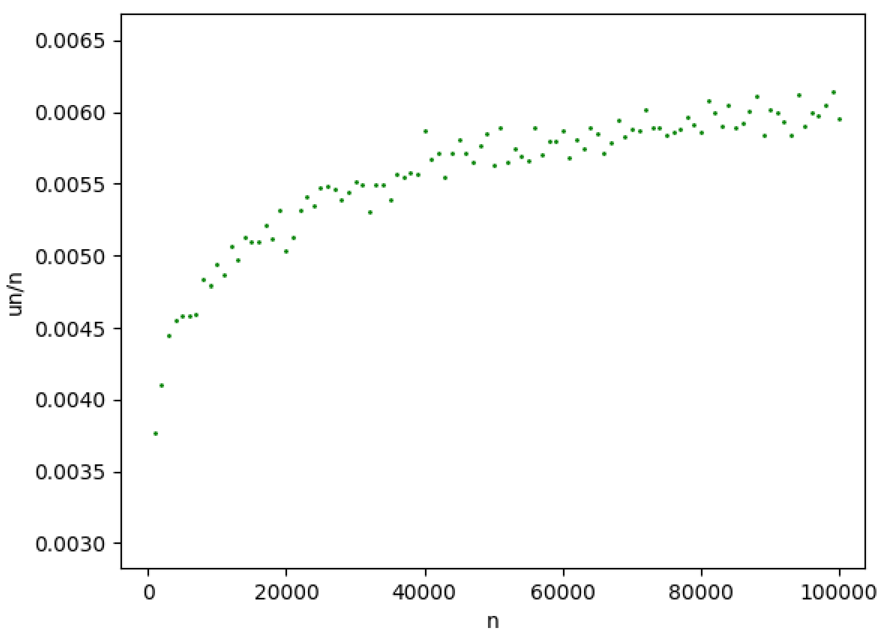



Wykres c(n)/n - niebieskie punkty , c(n)/n/ln(n) - czerwone punkty, c(n)/n/n - zielone punkty.




Z wykresu ciężko jest wywnioskować, które punkty bardziej przypominają funkcję stałą. Po przybliżeniu widać jednak, że punkty zielone mieszczą się w dużo mniejszym zakresie.



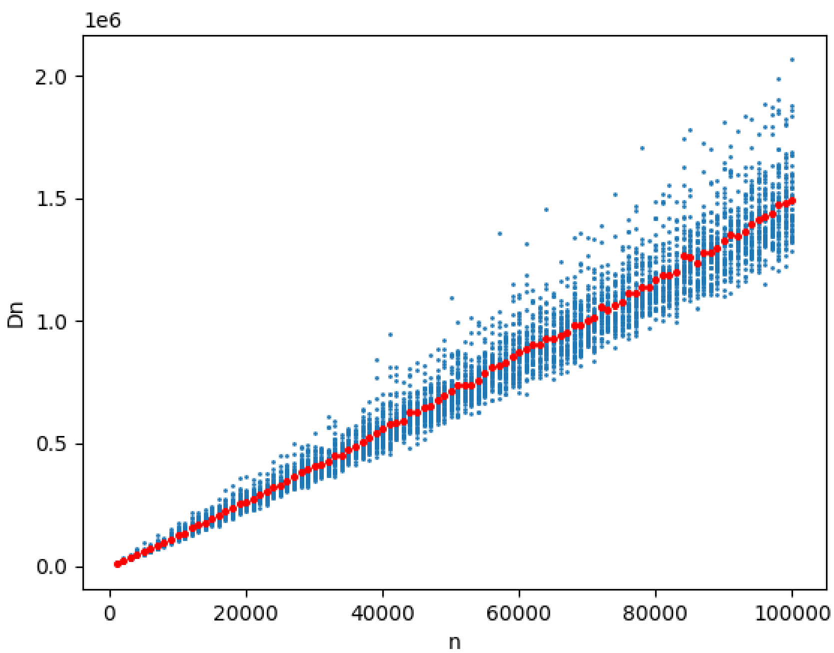




To oznacza, że Cn = O(n*n)

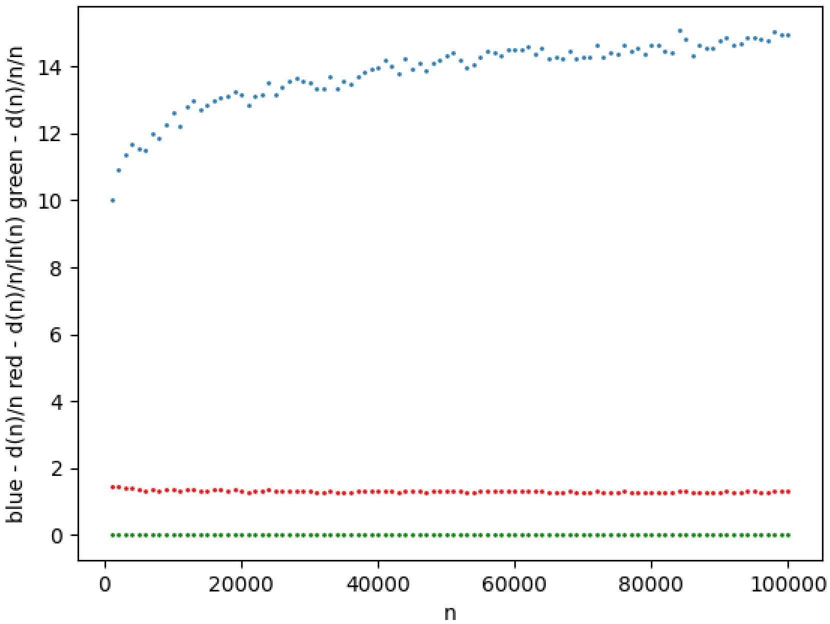
Ten podpunkt zawiera w sobie coupon collector’s problem. Problem kolekcjonera kuponów opisuje klasę konkursów, w którym gracz otrzymuje wygraną po zebraniu wszystkich kuponów z określonej puli. Problem polega na przewidzeniu jak długo należy zbierać kupony, aby otrzymać wygraną. W naszym przypadku pula kuponów to zbiór urn, a wygraną jest moment Cn.

Źródło: wikipedia.org

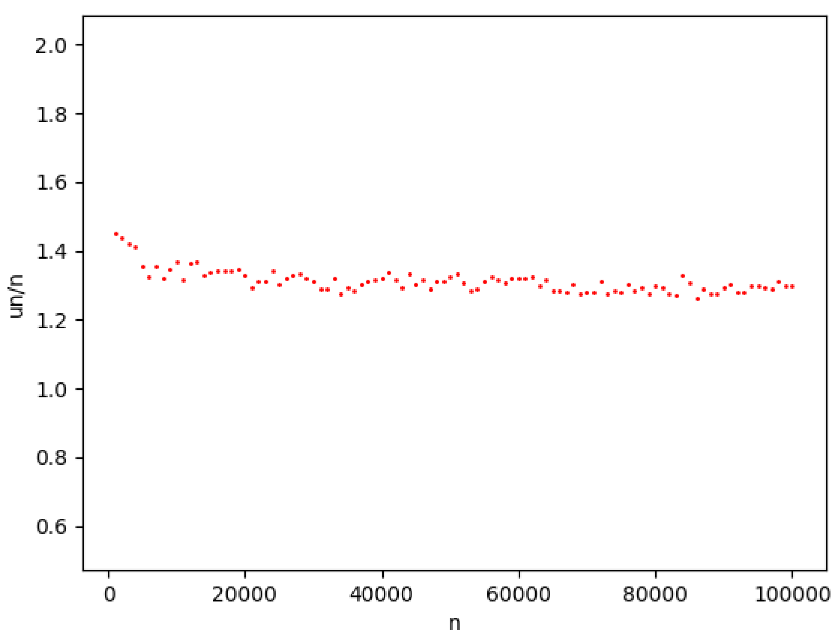
e) Dn – minimalna liczba rzutów, po której w każdej z urn są co najmniej dwie kule. Koncentracja punktów wokół wartości średniej słabnie wraz ze wzrostem n. Wartości średnie sugerują funkcję liniową ale zmieni się to na drugim wykresie.



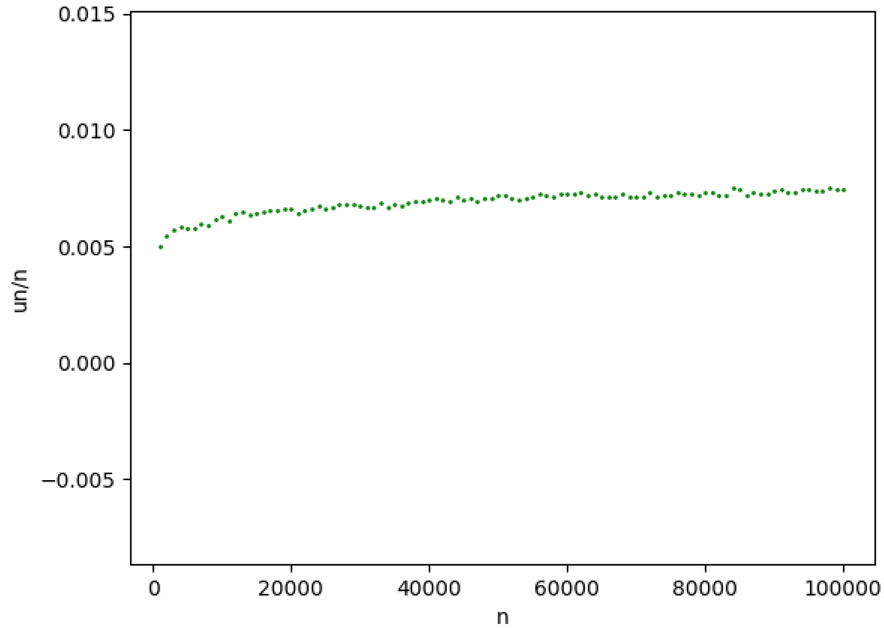

Wykres d(n)/n - punkty niebieskie, d(n)/n/ln(n) - punkty czerwone oraz d(n)/n/n - punkty zielone. Po przybliżeniu ponownie widać, że punkty zielone mieszczą się w mniejszym zakresie niż punkty czerwone .



blue - d(n)/n red - d(n)/n/ln(n) green - d(n)/n/n
0
2
4
6
8
10
12
14
0
20000
40000
60000
80000
100000
n

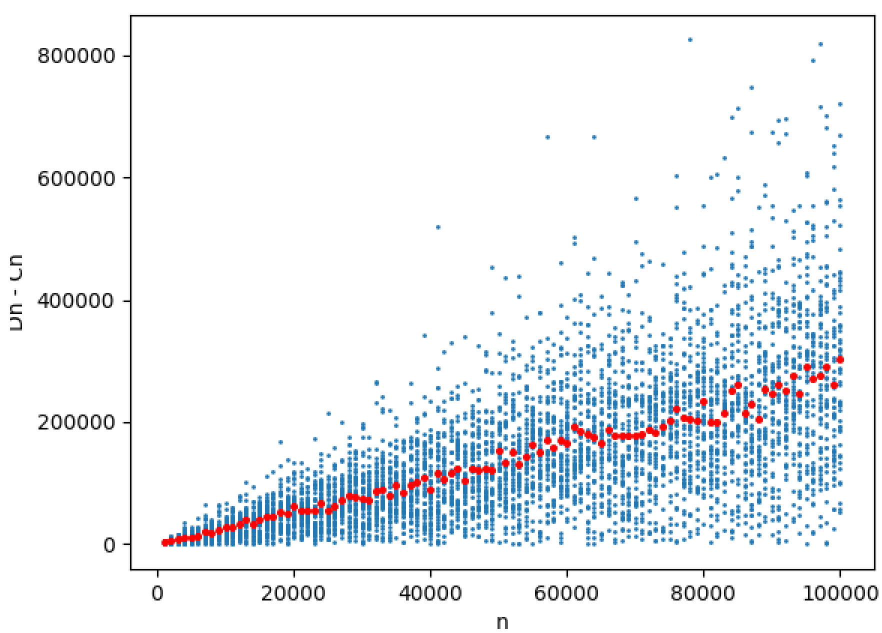



un/n
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
0
20000
40000
60000
80000
100000
n

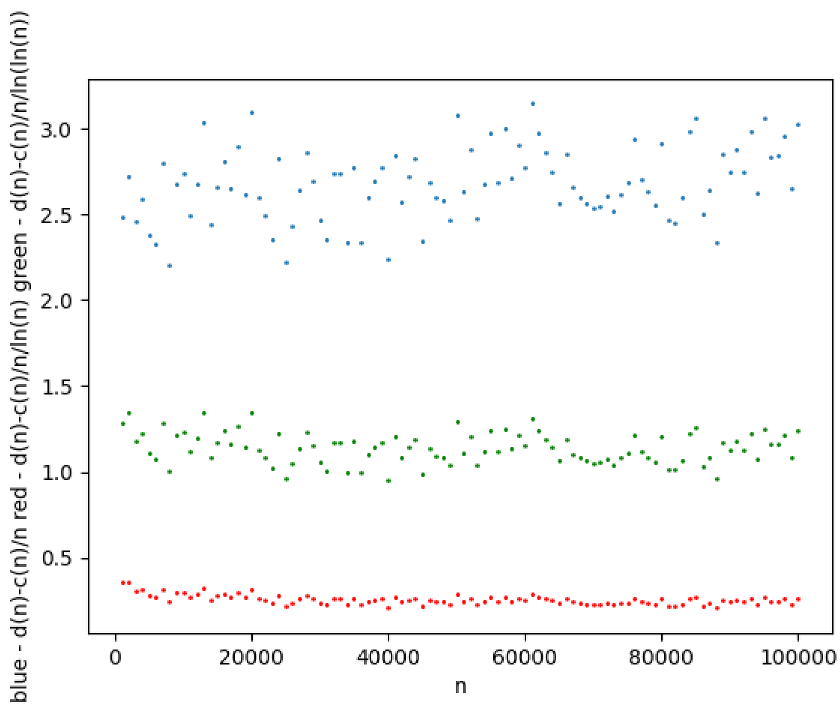



To oznacza, że w przybliżeniu Dn = O(n*n)

f) Dn – Cn -liczba rzutów od momentu Cn potrzebna do tego, żeby w każdej urnie były co najmniej dwie kule. Koncentracja punktów wokół wartości średnich słabnie wraz ze wzrostem n.



Wykres (d(n)-c(n))/n - punkty niebieskie , (d(n)-c(n))/n/ln(n) - punkty czerwone , (d(n)-c(n))/n/ln(ln(n)) - punkty zielone. Punkty czerwone najbardziej przypominają funkcję stałą. Można stąd wywnioskować, że w przybliżeniu Dn-Cn = O(n*ln(ln((n)))




Dziękuję za uwagę.