MPiS Zadanie domowe 1

Manfred Gawlas

Parametry symulacji

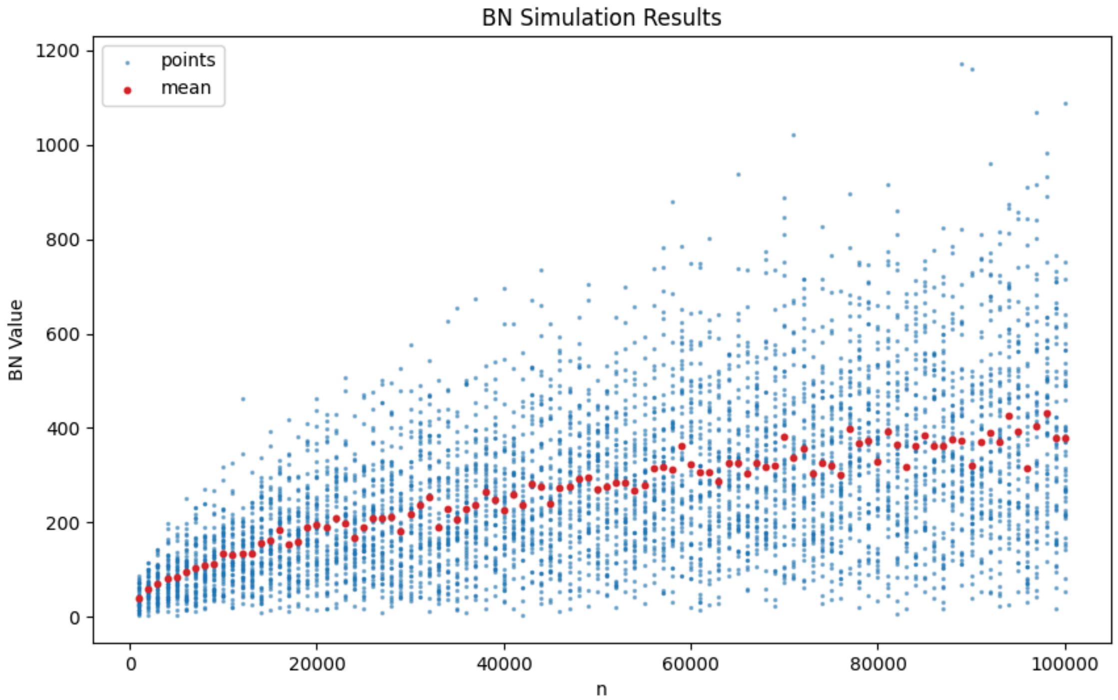
Do symulacji wykorzystano generator Mersenne Twister, którego implementacja pochodzi z Wikipedii. Seed został ustalony, zgodnie z polecieniem z Wikipedii, na:

```
uint32_t seed = 19650218UL;
```

Wyniki dla wielkości z widocznym rozkładem

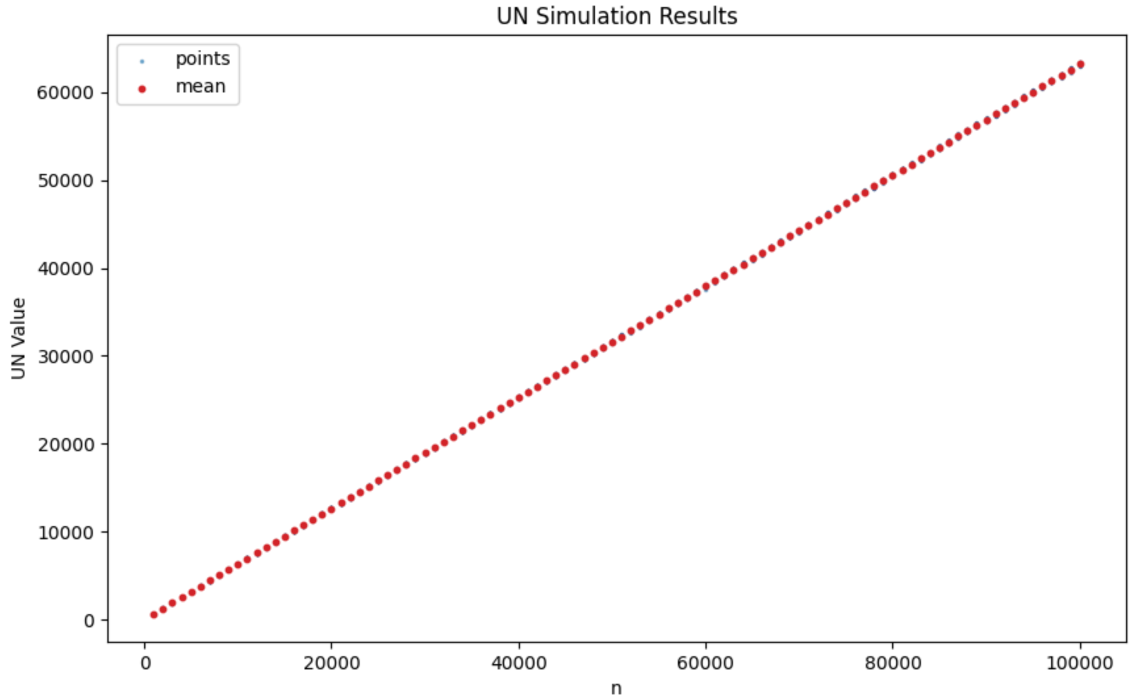
Bn

Wartość Bn jest bez zaskoczenia niska dla wszystkich wartości oraz rozkład jest dosyć duży, gdzie wśród poszczególnych symulacji widzimy rozrzut nawet do 3 razy wartości średniej Bn.



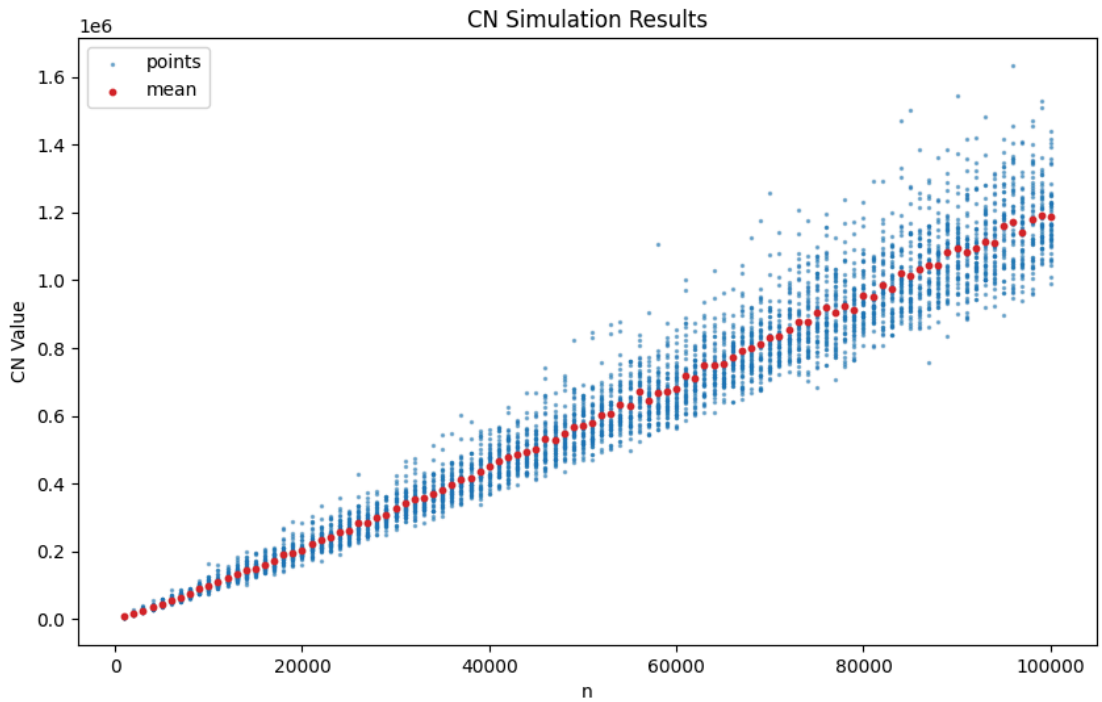
Un

Wartość Un rośnie idealnie liniowo i rozrzót jest tak mały że nawet go nie widać. Liniowość wzrostu nie jest zaskoczeniem.



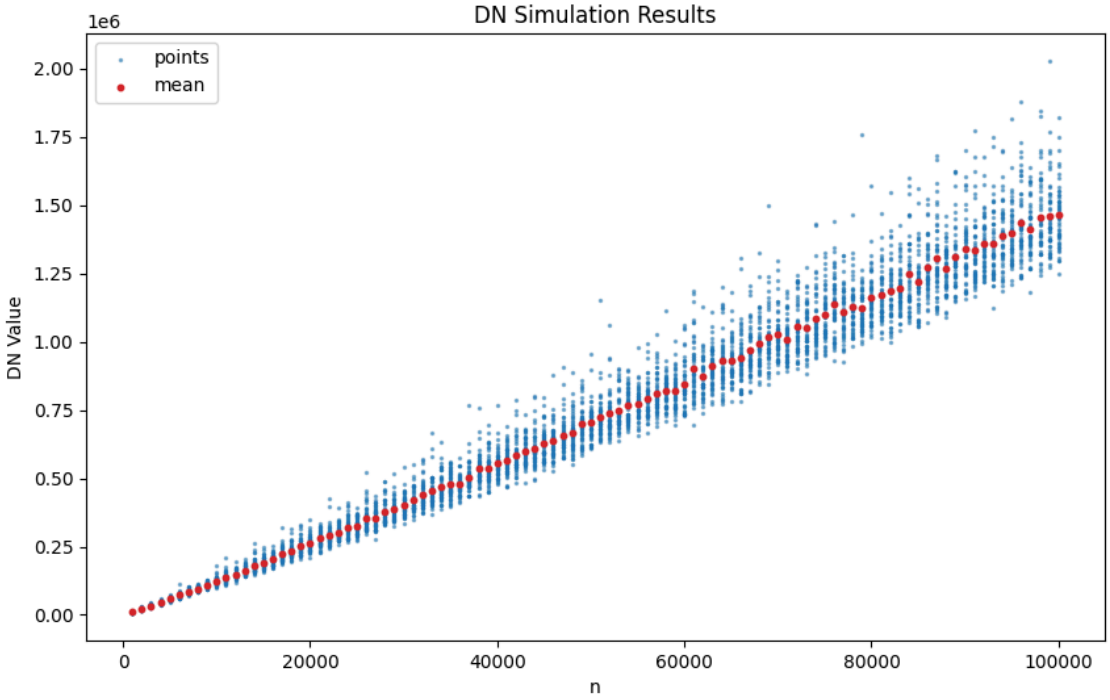
Cn

Cn zdaje się z tego wykresu rosnąć wmiare liniowo, wartość odchylenia standardowego wzrasta ze wzrostem n.



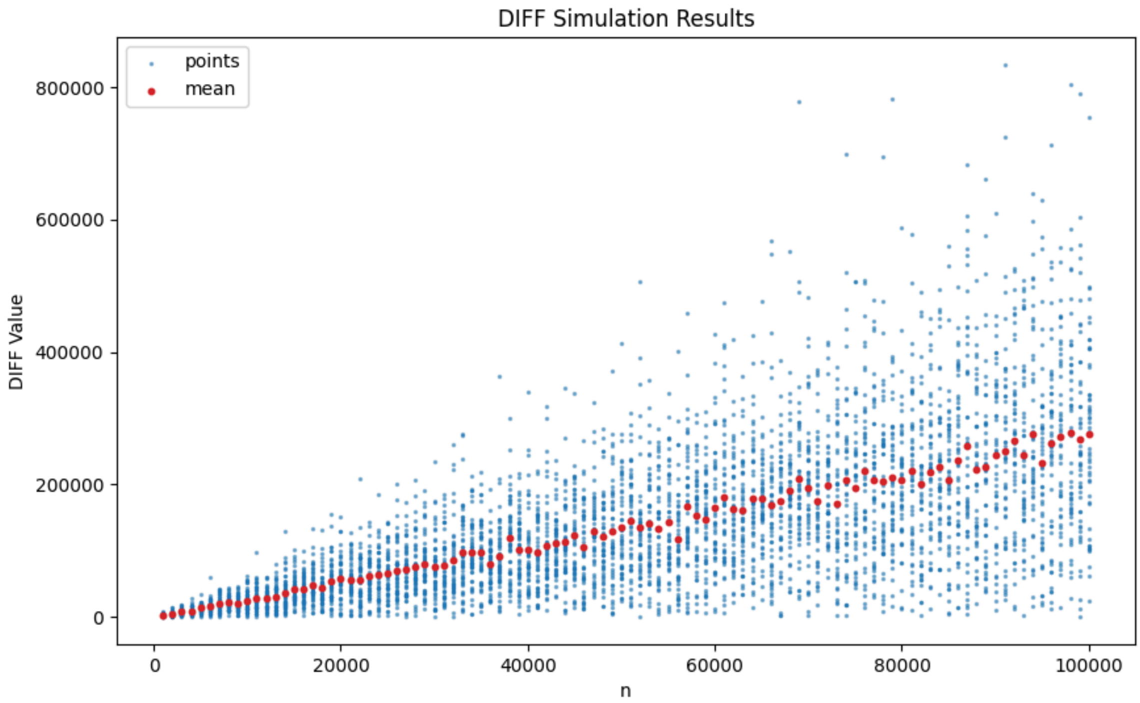
Dn

Dn zachowóje sie podobnie do Cn.



(Dn - Cn)

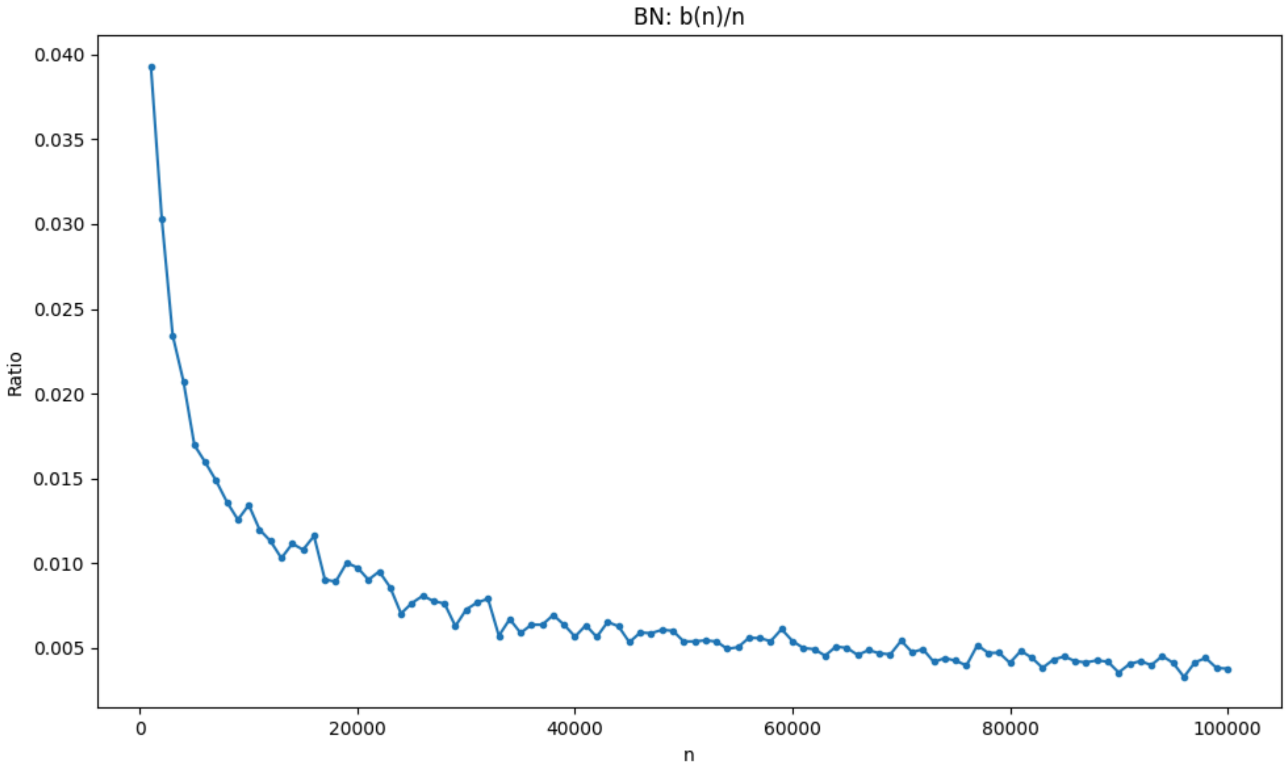
Zdaje się być funkcją liniową, ponownie odchylenie standardowe wzrasta wraz z n, ale jest zdecydowanie większe niż w obu poprzednich wykresach.

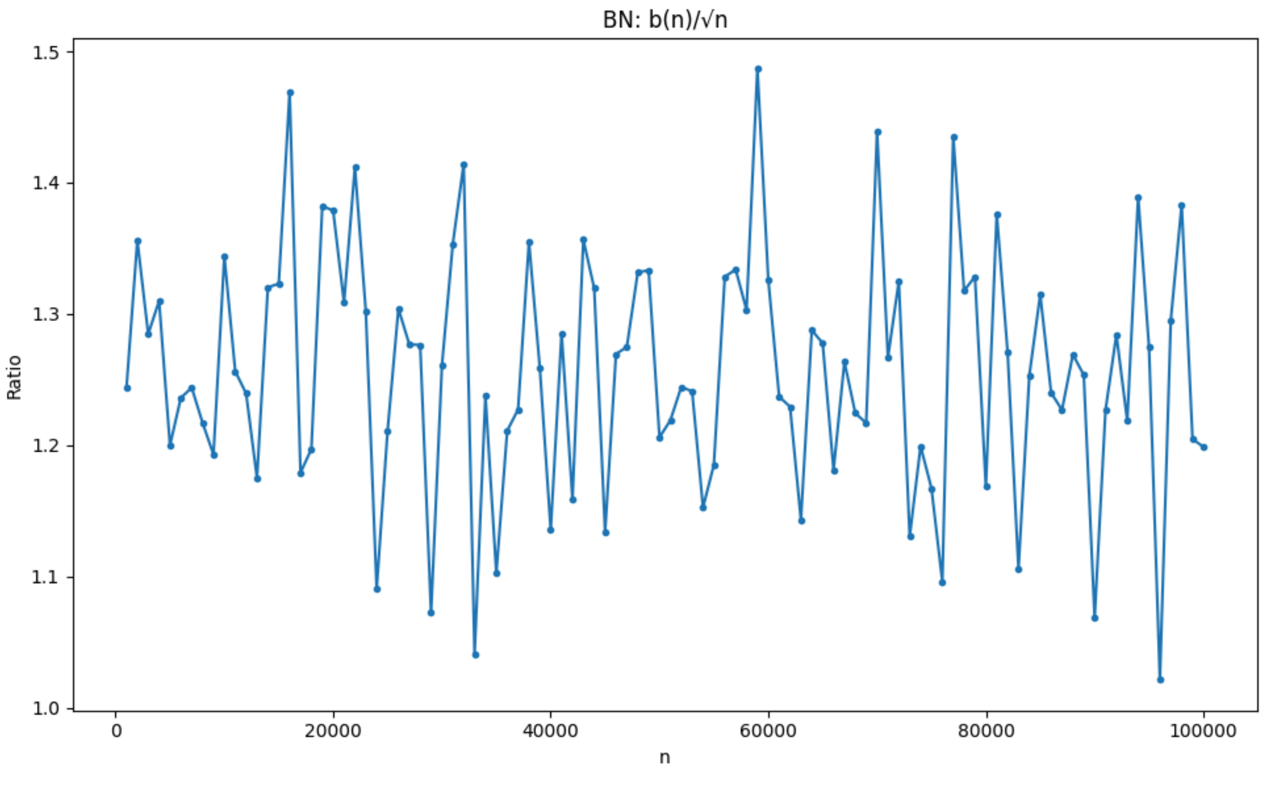


Wykresy stosunków funkcji z badań oraz funkcji z zadania

Bn

Wykresy te jednoznacznie wskazują, pomimo szumu że asymptota bn to $O(\sqrt{n})$.



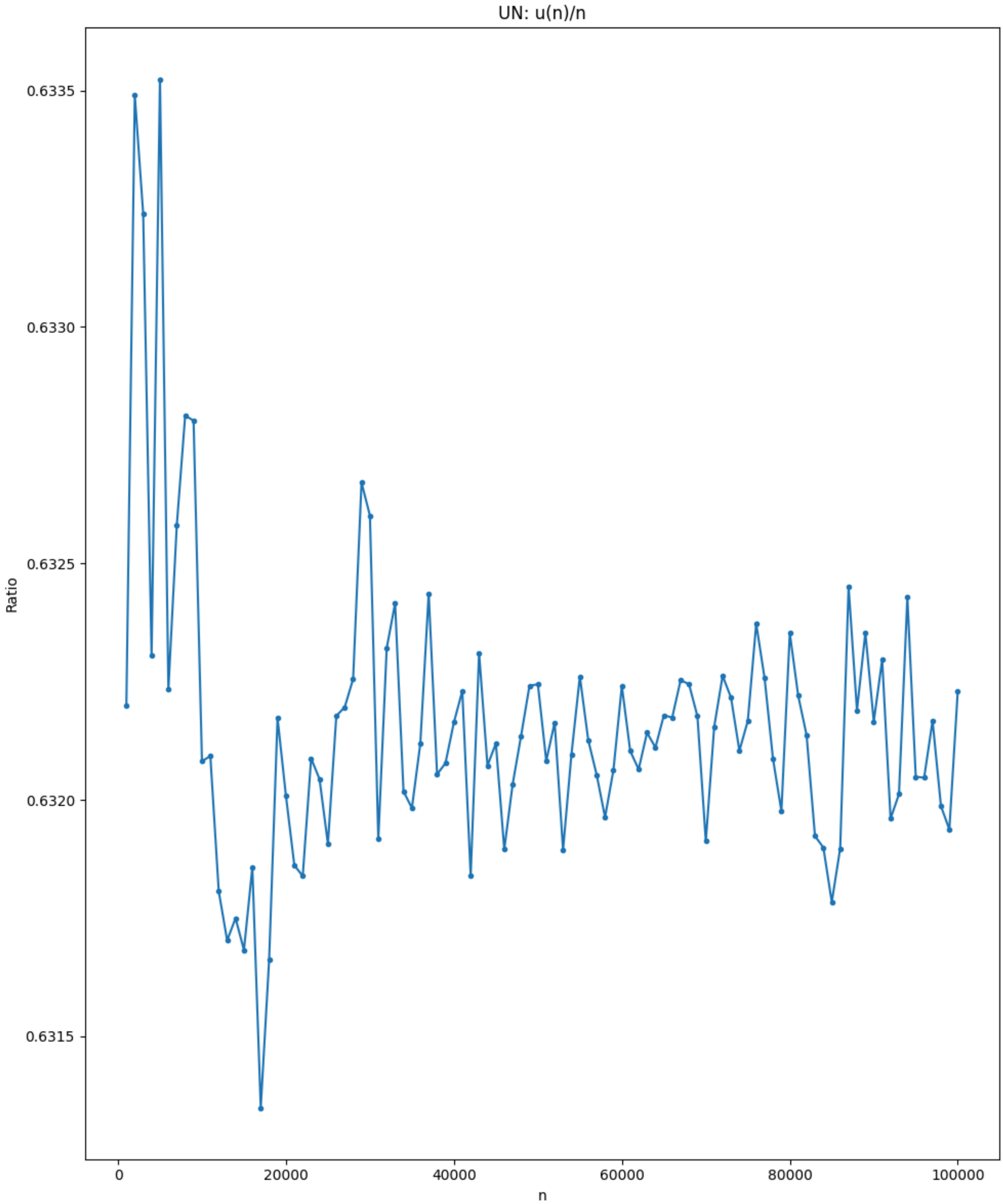


Un

Asymptota u(n):

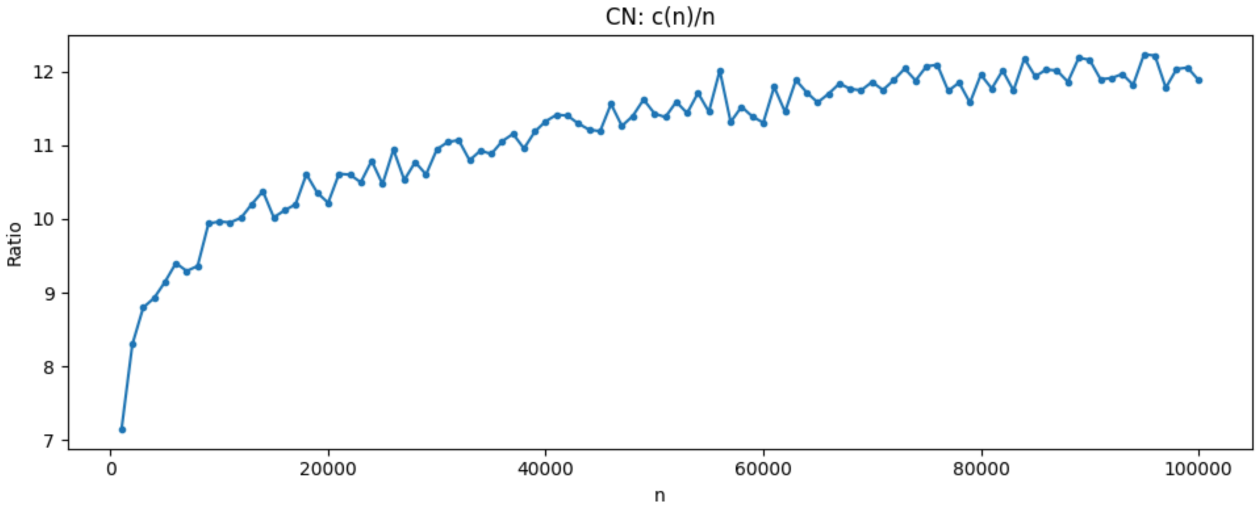
$$O(n)$$

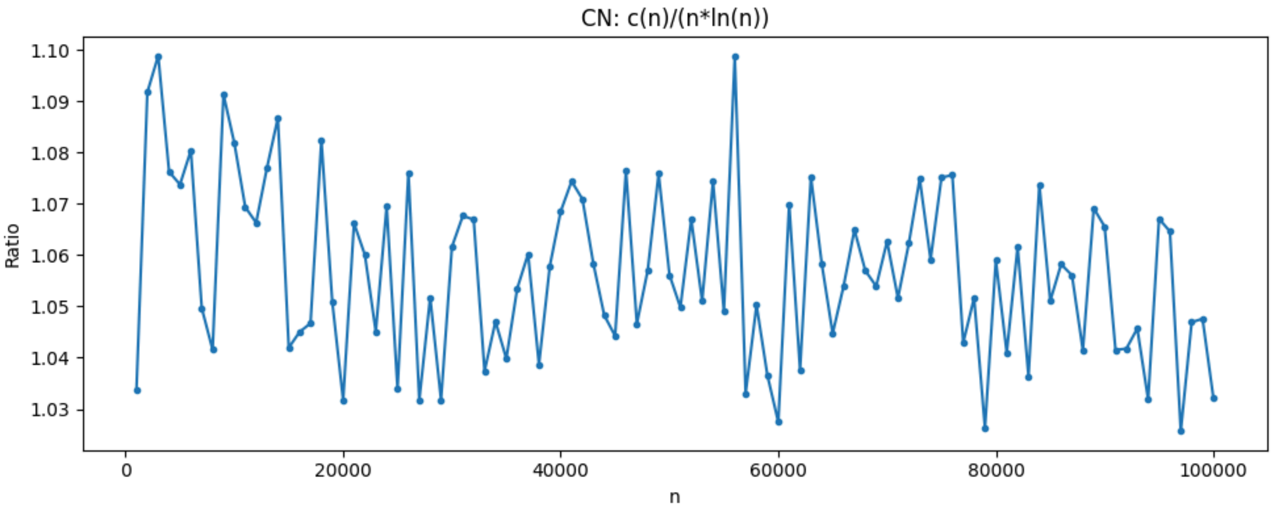
Relatywny peak na początku jest nadal bardzo mały patrząc na wartości liczbowe.

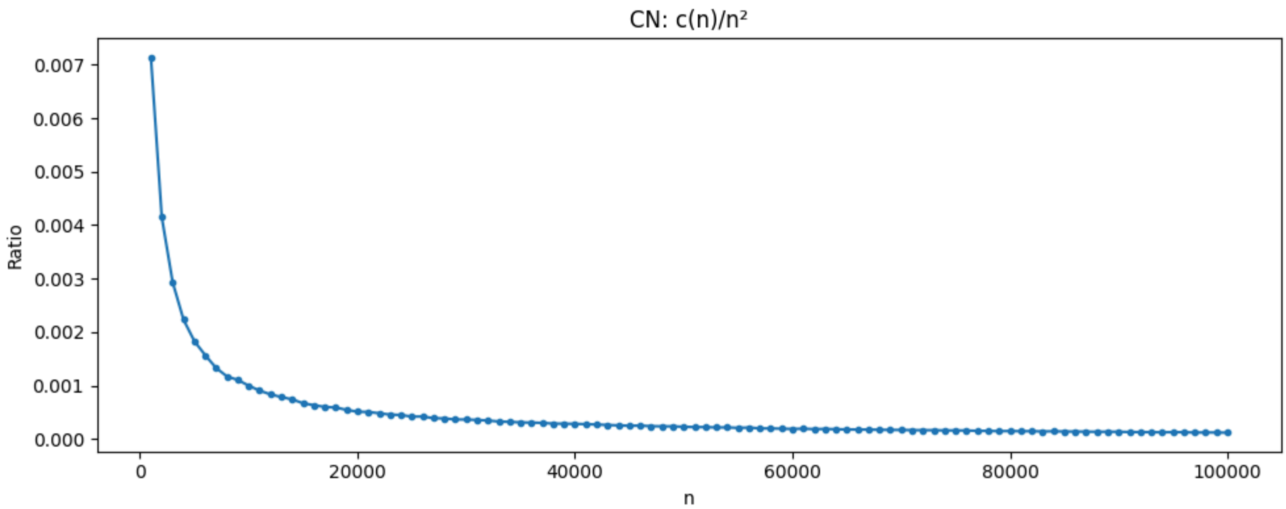


Cn Ratio Plots

Asymptota ponownie zdaje się być dokładnie $O(n \ln n)$

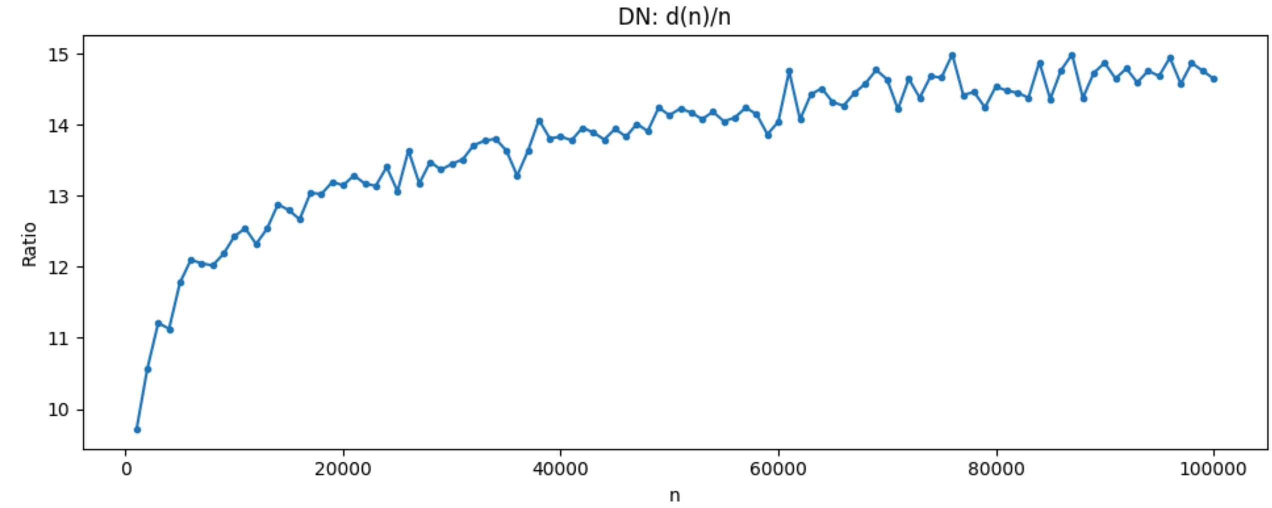


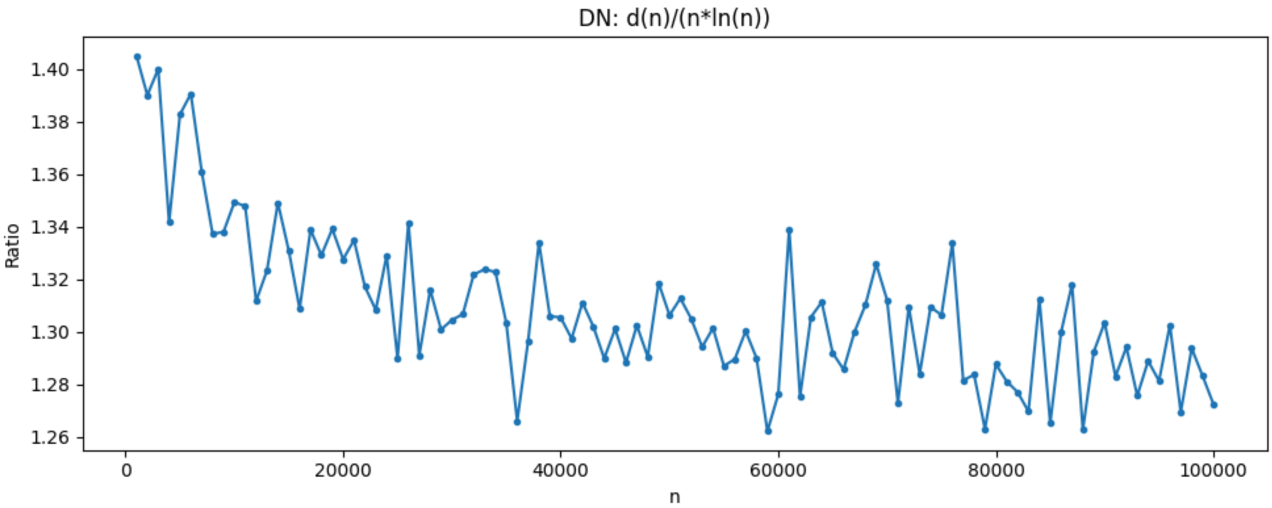


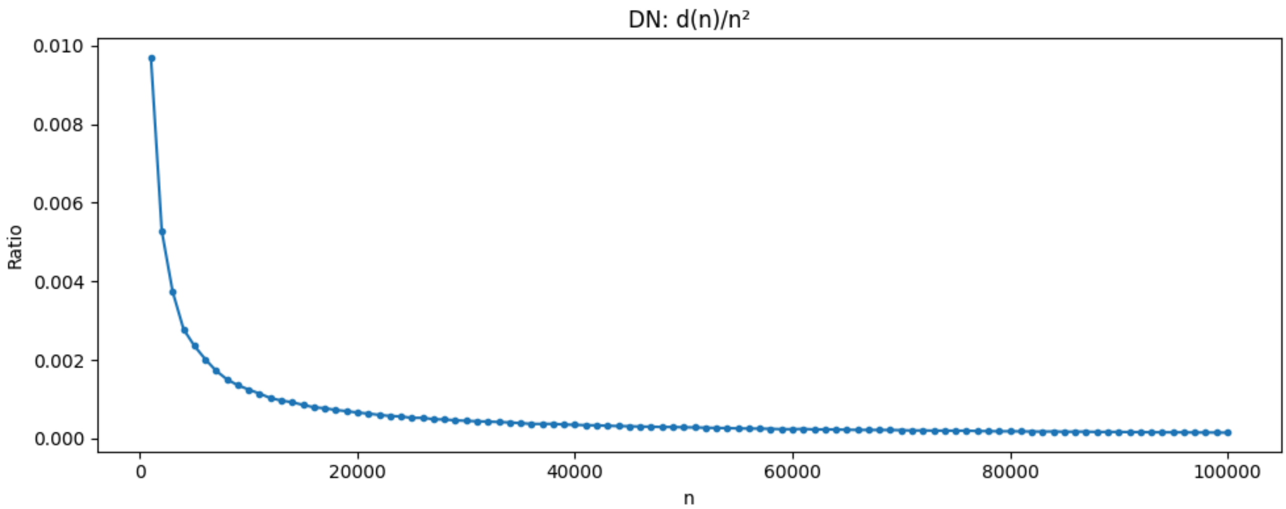


Dn Ratio Plots

W tym przypadku najlepszym przybliżeniem z wykonanych wykresów jest $O(n \ln n)$

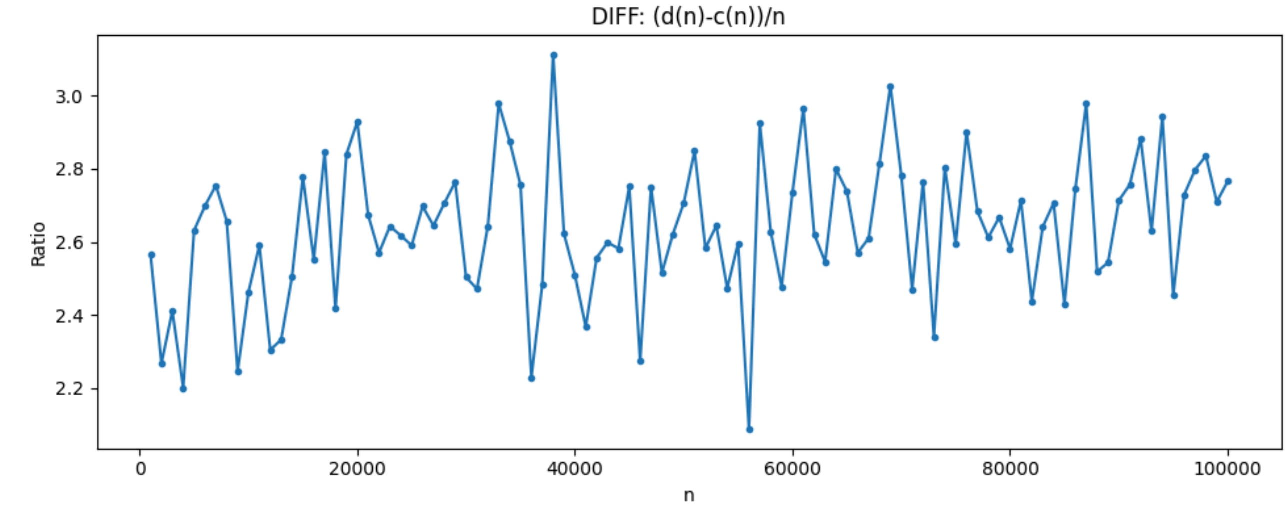


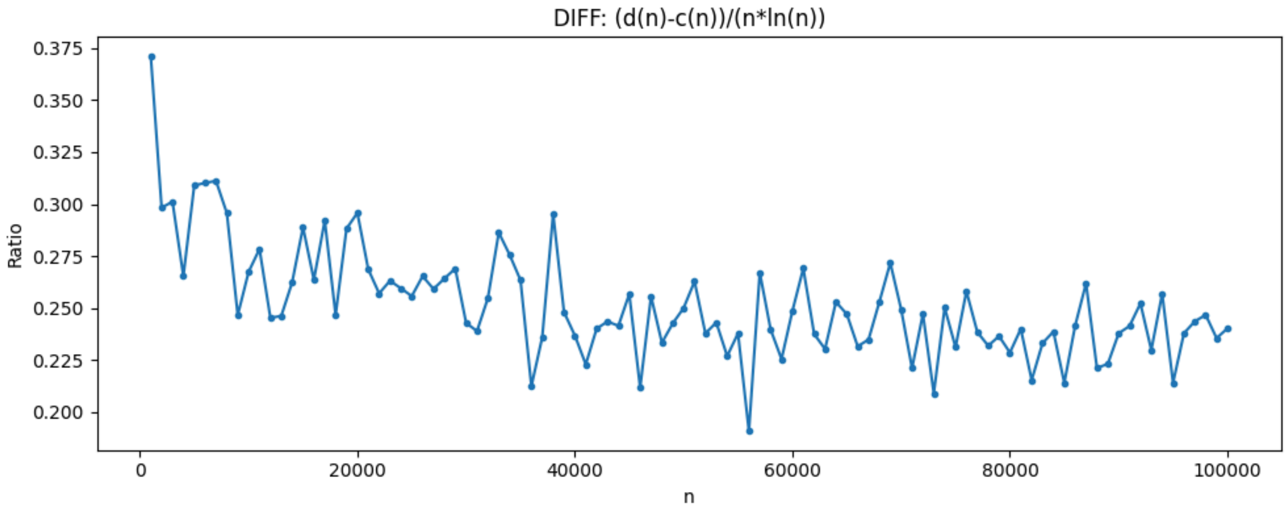


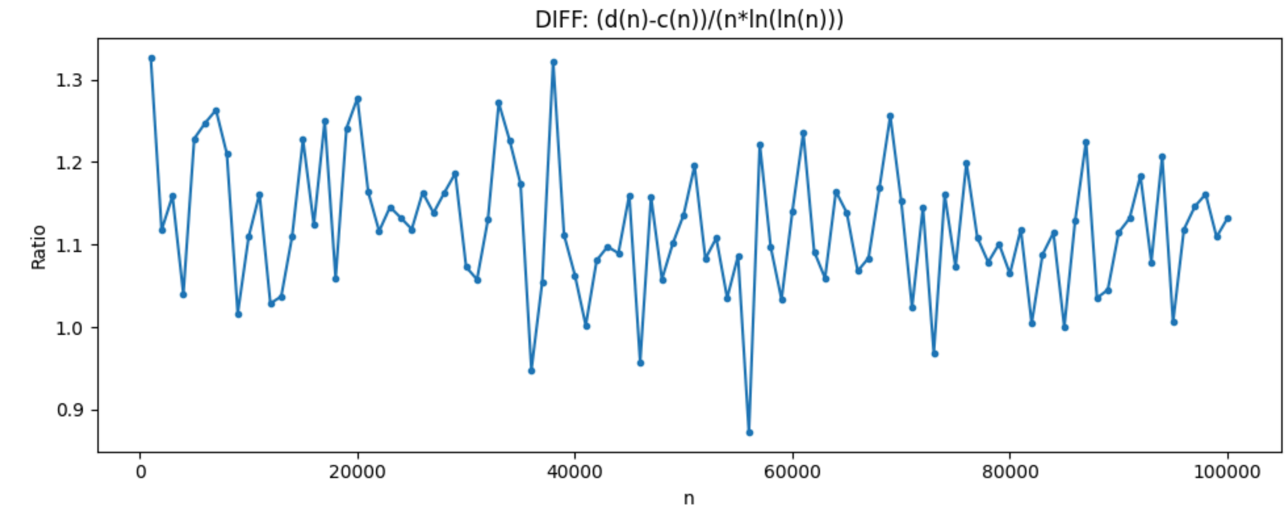


Difference (Dn - Cn) Ratio Plots

Tutaj co ciekawe asymptotyka tej funkcji to $O(n \ln(\ln n))$







Birthday paradox

Użycie nazwy birhday paradox co do funkcji b(n) jest uzasadnione dlatego, że możemy pomyśleć o kubłach jako o dniach i kulkach jako o losowo wybranych ludziach. b(n) mówi nam kiedy(a więc przy jakiej ilości ludzi) pierwszy raz wydaży się sytuacja gdzie dwie kule są w tym samym kuble(dwójka ludzi ma urodziny w ten sam dzień).

A więc gdyby założyć n = 365 to będzie praktycznie ta sama sytuacja co zakładana w birthday paradox.

Coupon colector problem

Zadanie Cn jest dosłownie tym problemem, dlatego że problem pyta się ile trzeba kupić produktów z losowym kuponem(wylosować kulek) by mieć po jednej sztuce każdego z n rodzajów kuponu(każdy z n kubków ma 1 kulke).

Tutaj nasza odpowiedź na temat asymptotyki tego problemu pokrywa się z tą na temat coupon colector problem.

Znaczenie birthday paradox'u w funkcjach hashujących

Funkcje hashujące, w szczególności jakieś nie trywialne, można w pewnym stopniu nazwać deterministyczną funkcją na pseudo-losową liczbę(statystycznie wyglądającą na pseudo-losową). Tak jak więc w tym naszym przykładzie, powiedzmy że mamy 1000 elementową przestrzeń hash'y. Dla różnych wartości, funkcja hashująca(w szczególności kryptograficzne funkcje hashujące) będzie przeżucać nasz input na pewną względnie pseudo losową liczbę z tego zakresu. Tak jak wiemy z przykładu i z birthday paradox'u, po $\sqrt{n}$ przeżuceniach dostaniemy stytuacje w której dwóm różnym obiektą-wartością będzie odpowiadać ten sam hash(kubek z naszego zadania).

Tak więc birthday paradox ma duże znaczenie w zagadnieniu funkcji hashujących i kryptograficznych funkcji hashującyh, gdyż ogranicza nam maksymalną ilość wartości które możemy sensownie z małym prawdopodobieństwem powtórzeń zhashować dla n elementowej przestrzeni hashy do pierwiastka z n.