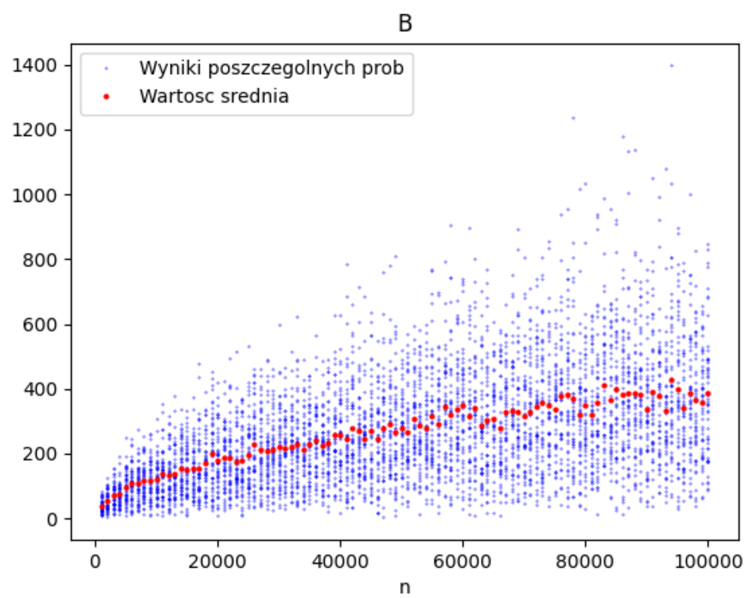
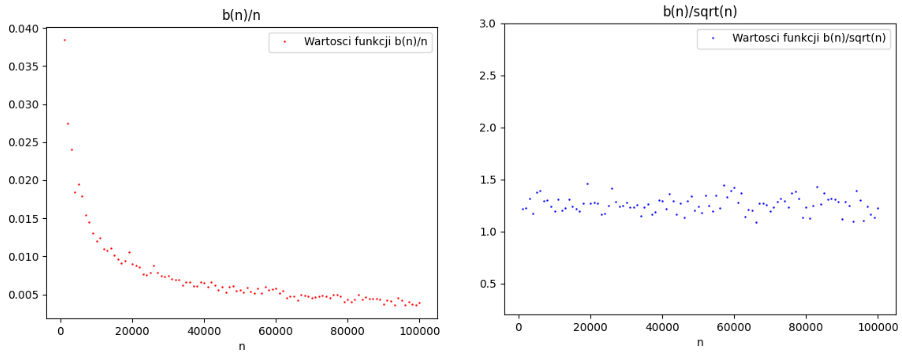
Moment pierwszej kolizji (B_n)

Moment pierwszej kolizji to punkt w naszej symulacji gdy jakiś kubeł został wybrany po raz drugi. Jest to wariacja problemu Birthday Paradox, który opisuje szansę na to, że dwie osoby w m-osobowej grupie mają urodziny w ten sam dzień. Paradoks polega na tym, że szansa na to aby dwie osoby współdzieliły dzień urodzin wynosi 50% już dla tylko 23 osób. Zmienna B_n symuluje birthday paradox dla $n = 365$ co widzimy także na wykresie (dla n tak dużego jak 100000 potrzebujemy średnio zaledwie około 400 prób). Na wykresie widzimy, że koncentracja wokół średniej jest dość niska oraz rośnie wraz z n.



Rysunek 1: Wykres przedstawiający wyniki poszczególnych eksperymentów oraz wartość średnią B_n



(a) $\frac{B_n}{n}$ (b) $\frac{B_n}{\sqrt{n}}$

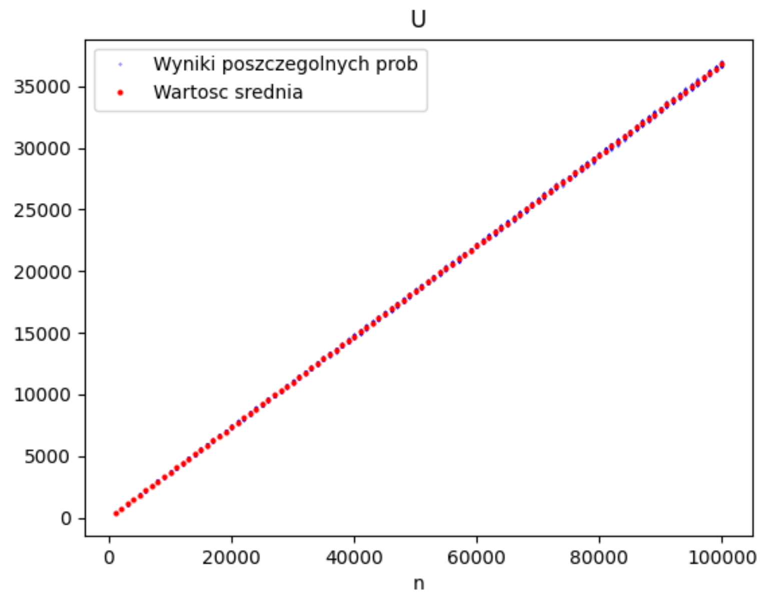
Rysunek 2: Wykresy pomagające w znalezieniu asymptotyki funkcji B_n

Hipoteza:

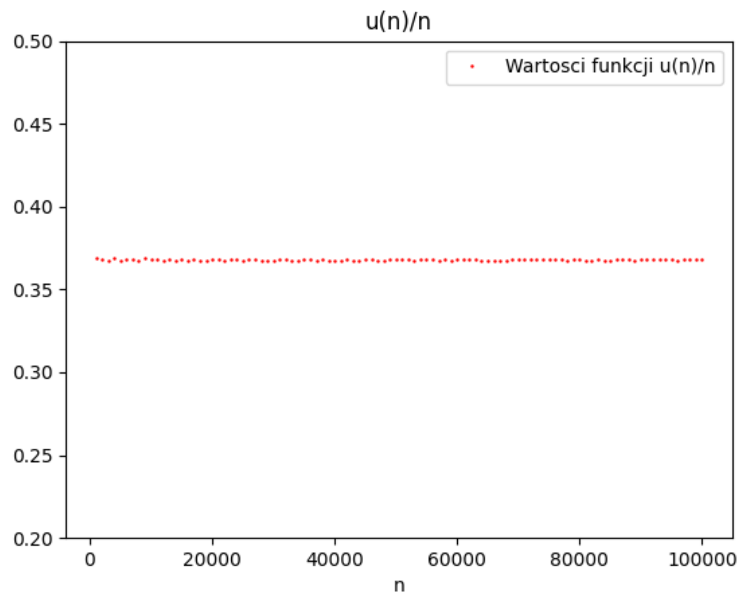
$$B_n = O(\sqrt{n})$$

Liczba pustych urn po wrzuceniu n kul (U_n)

Ilość pustych urn po wrzuceniu n kul. Obserwacje są silnie skoncentrowane wokół wartości średniej, ponadto ich koncentracja nie rośnie wraz z n.



Rysunek 3: Wykres przedstawiający wyniki poszczególnych eksperymentów oraz wartość średnią U_n



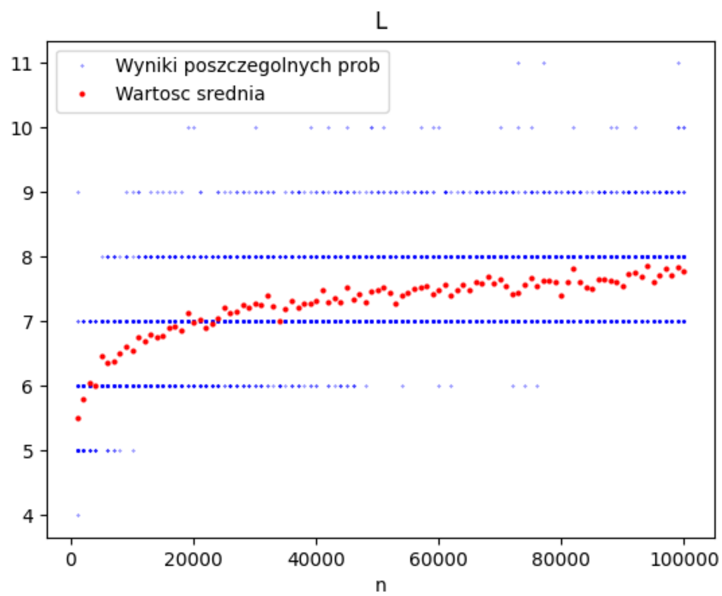
Rysunek 4: Wykres pomagający w znalezieniu asymptotyki funkcji U_n

Hipoteza:

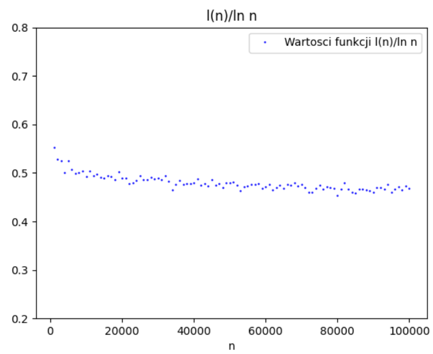
$$U_n = O(n)$$

Maximum load (L_n)

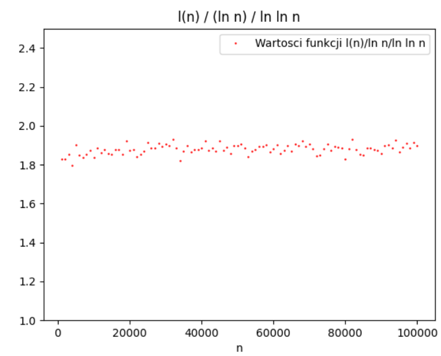
Obserwacje są względnie silnie skoncentrowane wokół wartości średniej, nie odbiegają one od niej o więcej niż 4 i zamykają w przedziale $[4, 11]$.



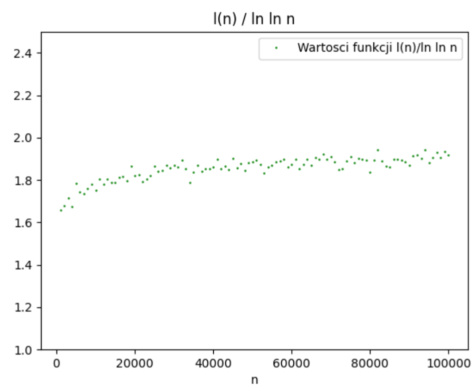
Rysunek 5: Wykres przedstawiający wyniki poszczególnych eksperymentów oraz wartość średnią L_n



(a) $\frac{L_n}{ln(n)}$



(b) $\frac{L_n}{\frac{ln(n)}{ln(ln(n))}}$



(c) $\frac{L_n}{ln(ln(n))}$

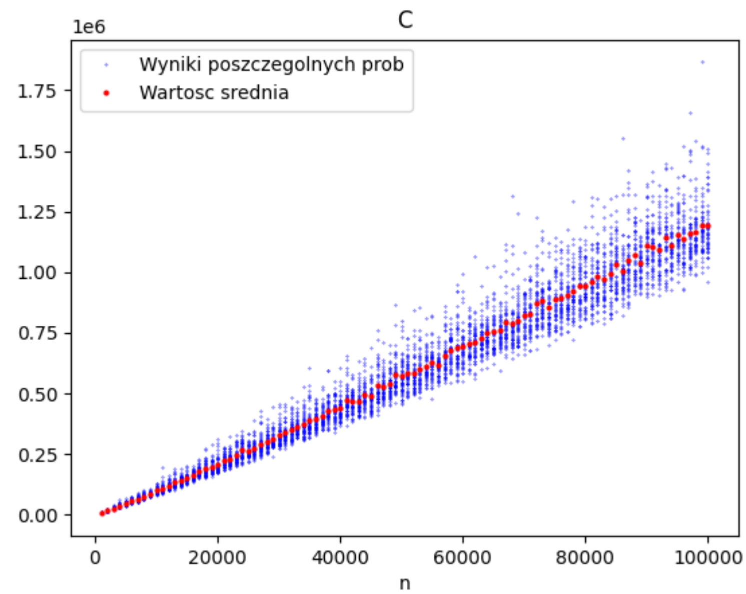
Rysunek 6: Wykresy pomagające w znalezieniu asymptotyki funkcji L_n

Hipoteza:

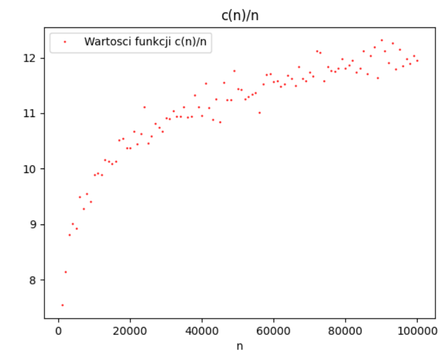
$$L_n = O(\frac{ln(n)}{ln(ln(n))})$$

Pierwszy moment, po którym nie ma już pustych urn (C_n)

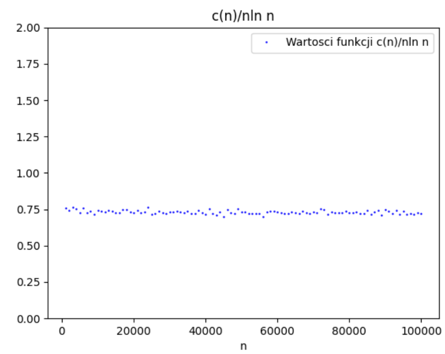
Minimalna liczba rzutów, po której w każdej z urn jest co najmniej jedna kula. Zmienna ta modeluje Coupon collector's problem, który opsiuje ile razy musimy kupić kupon, aby zdobyć wszystkie możliwe z kolekcji n różnych. Jest to analogiczne do naszej symulacji (ile razy losujemy kubeł żeby we wszystkich była co najmniej jedna piłka). Koncentracja wokół średniej początkowo jest dość silna, ale rośnie ona wraz z wartością n.



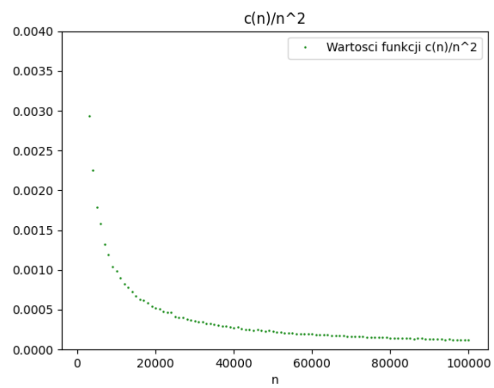
Rysunek 7: Wykres przedstawiający wyniki poszczególnych eksperymentów oraz wartość średnią C_n



(a) $\frac{C_n}{n}$



(b) $\frac{C_n}{nln(n)}$



(c) $\frac{C_n}{n^2}$

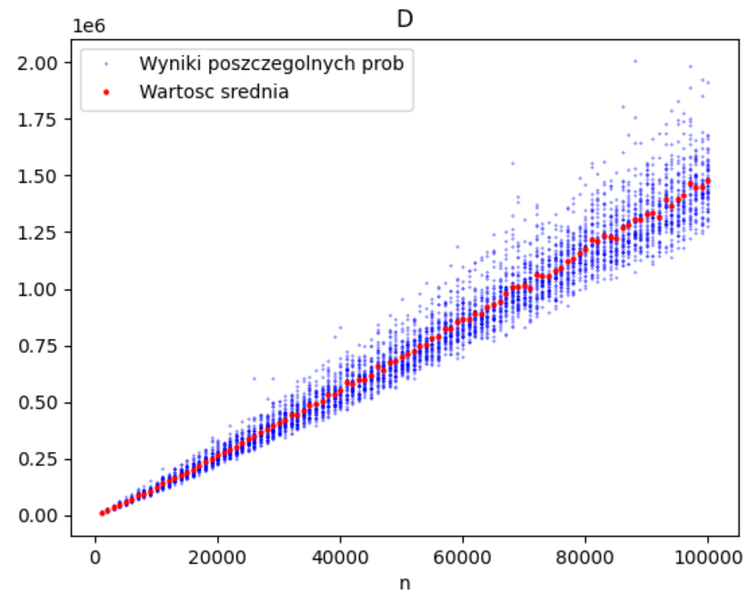
Rysunek 8: Wykresy pomagające w znalezieniu asymptotyki funkcji C_n

Hipoteza:

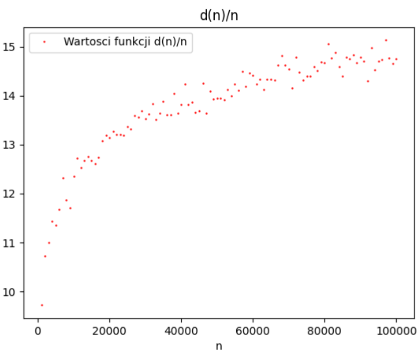
$$C_n = O(nln(n))$$

Minimalna liczba rzutów, po której w kazdej z urn są co najmniej dwie kule (D_n)

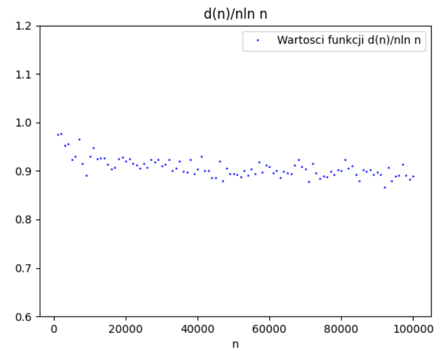
Tak jak wyżej koncetracja średniej początkowo jest dość silna, ale rośnie wraz z wartością n.



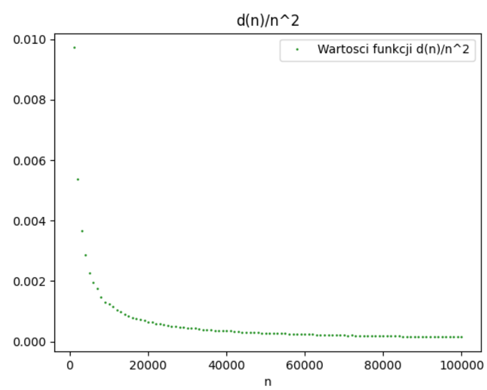
Rysunek 9: Wykres przedstawiający wyniki poszczególnych eksperymentów oraz wartość średnią D_n



(a) $\frac{D_n}{n}$



(b) $\frac{D_n}{nln(n)}$



(c) $\frac{D_n}{n^2}$

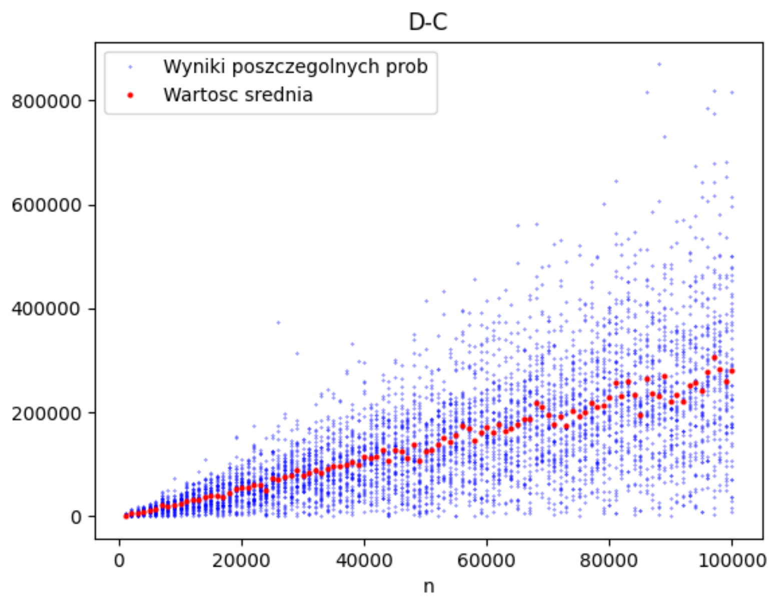
Rysunek 10: Wykresy pomagające w znalezieniu asymptotyki funkcji D_n

Hipoteza:

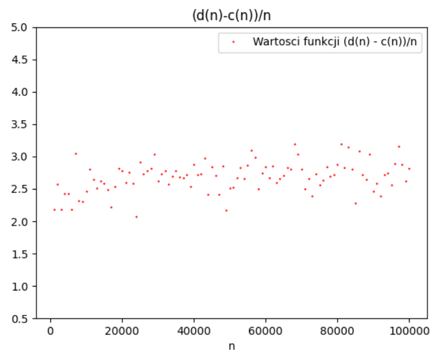
$$D_n = O(nln(n))$$

Różnica $(D_n - C_n)$

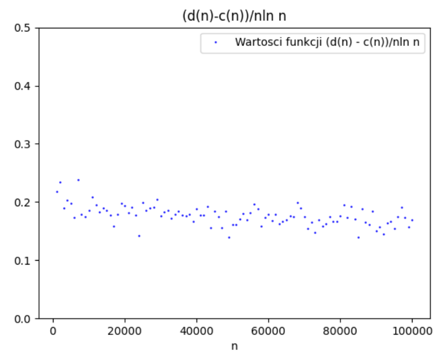
Ilość rzutów potrzebna do przejścia od momentu C_n do momentu D_n. Tutaj również koncentracja początkowo jest dość silna, ale przedział przyjmowanych wartości bardzo szybko rośnie, więc koncentracja słabnie.



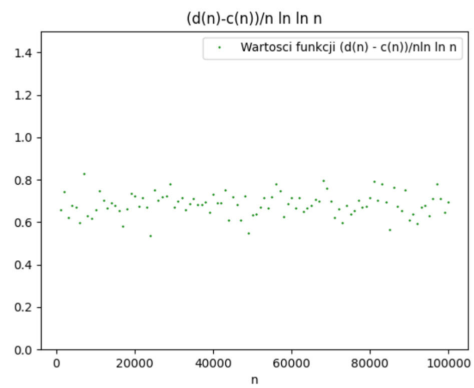
Rysunek 11: Wykres przedstawiający wyniki poszczególnych eksperymentów oraz wartość średnią $D_n - C_n$



(a) $\frac{D_n - C_n}{n}$



(b) $\frac{D_n - C_n}{nln(n)}$



(c) $\frac{D_n - C_n}{nln(ln(n))}$

Rysunek 12: Wykresy pomagające w znalezieniu asymptotyki funkcji $D_n - C_n$

W tym konkrentym przypadku nasz sample size jest zbyt mały żeby określić asymptotykę z samego wykresu. Obie funkcje D_n i C_n są asymptotyki $O(nln(n))$ więc $D_n - C_n \leqslant O(nln(n))$, ale wszystkie funkcje przedstawione na wykresach również spełniają tę własność.

$$D_n - C_n = O(?)$$

Birthday paradox w kontekście funkcji hashujących

Birthday paradox modeluje nam szansę, że nasz algorytm hashowania stworzy dwa identyczne hashe dla dwóch różnych wejść. Przede wszystkim pokazuje on, że pomimo tego, że nasza przestrzeń możliwych hashów jest duża, to szansa na to, że generując je wielokrotnie, dwa będą takie same jest stosunkowo niewielka. W dodatku maleje ona asymptotycznie do $O(\sqrt{n})$, więc dość wolno.