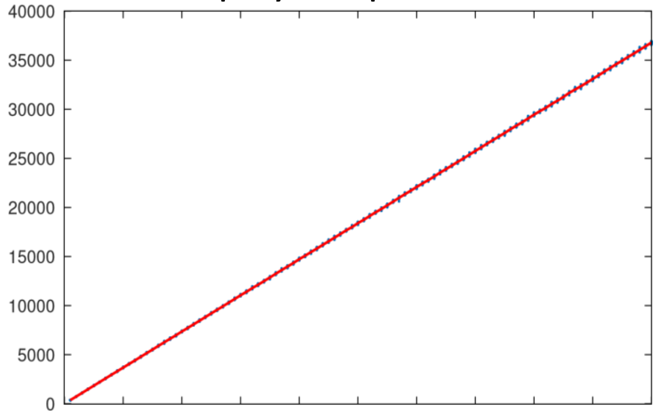
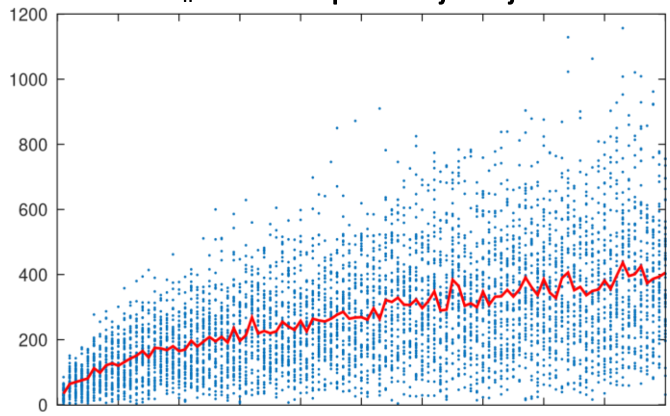
Zadanie domowe 2 – Nel Skowronek 279679

<u>Wykresy badanych wielkości z pierwszej części zadania:</u>

U_n – liczba pustych urn po wrzuceniu n kul



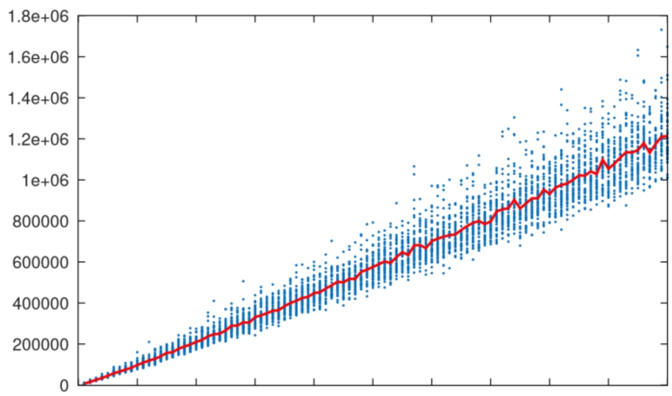
B_n – moment pierwszej kolizji



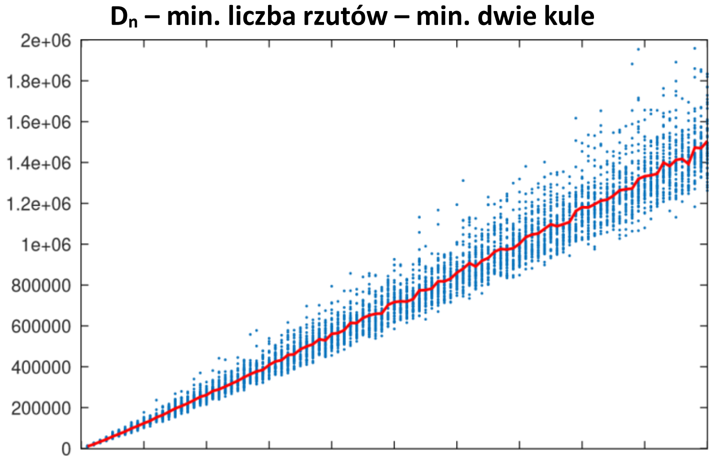
Jak widać najbardziej skoncentrowane wokół średniej są wyniki U_n, co sugeruje, że liczba pustych urn po n rzutach jest w dużej mierze przewidywalna i najprawdopodobniej zależy liniowo od n.

Wartości B_n są o wiele bardziej rozproszone, natomiast są bardzo małe w porównaniu do n. Oznacza to, że po pewnej małej liczbie rzutów prawdopodobieństwo, że wystąpiła kolizja jest bardzo duże – co nawiązuje do tego o czym mówi **birthday paradox** – już przy 23 osobach, prawdopodobieństwo, że dwie z nich mają urodziny w ten sam dzień, przekracza 50%. W naszej symulacji rolę osób odgrywają rzuty, a dni w roku – urny. (‘wrzucamy’ osoby w dzień urodzin)

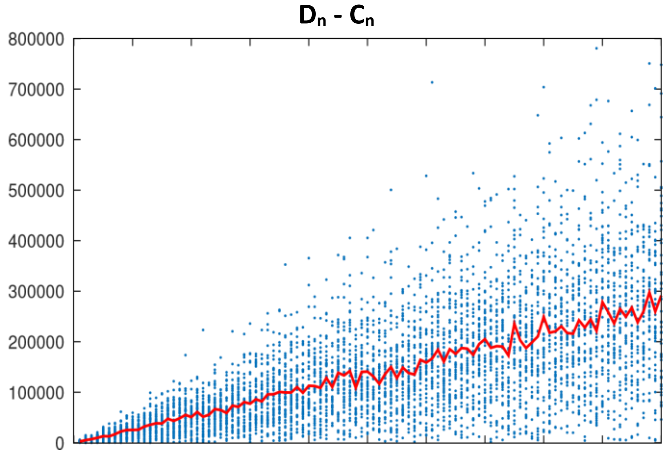
C_n – min. liczba rzutów – min. jedna kula



D_n – min. liczba rzutów – min. dwie kule



D_n - C_n



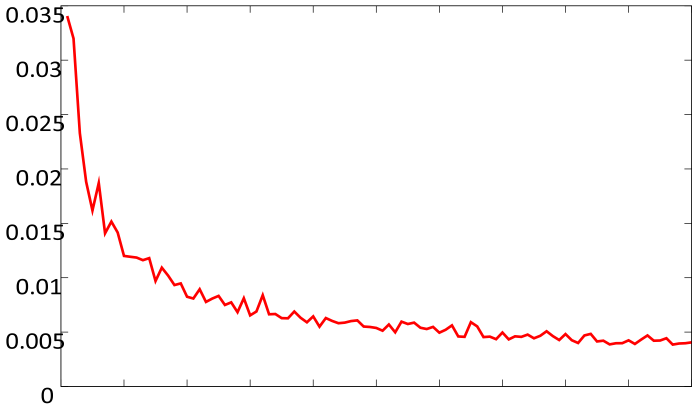
Wykresy C_n i D_n wyglądają bardzo podobnie – obydwa są w miarę skoncentrowane wokół wartości średniej i wydają się rosnąć nieco szybciej niż liniowo, D_n oczywiście szybciej niż C_n, prawdopodobnie o jakiś niższy rząd asymptotyczny. Wartości C_n i D_n reprezentują odpowiednio **coupon collector’s problem** i **coupon collector’s brother**, mierząc liczbę kupionych opakowań (wrzuconych kul), po której kolekcjoner wylosował po 1 (*2*) kuponie (urnie) każdego rodzaju.

D_n - C_n można intuicyjnie interpretować jako: „Jeśli wylosowałem każdą urnę (**kupon**) przynajmniej raz, to ‘dobić’ do 2 urn (**kuponów**) powinno być łatwiej, bo niektóre mam już wylosowane kilkukrotnie”.

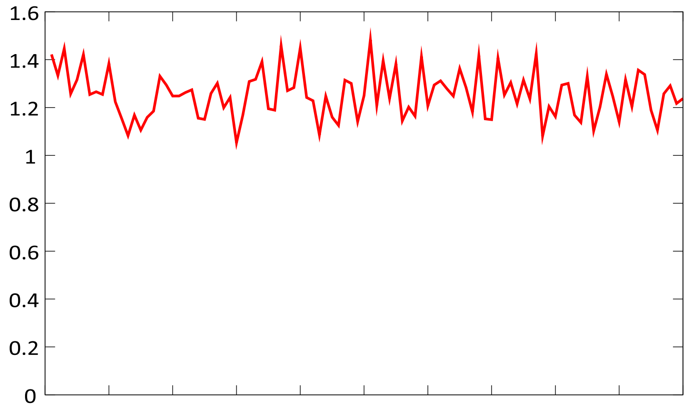
Oczywiście taka wartość będzie dużo mniejsza od C_n, co widać na wykresach. Rozproszenie wartości może wynikać z różniącego się momentu, od którego zaczynamy mierzyć ilczbę kul do ‘dobicia’.

Wykresy funkcji z drugiej części zadania:

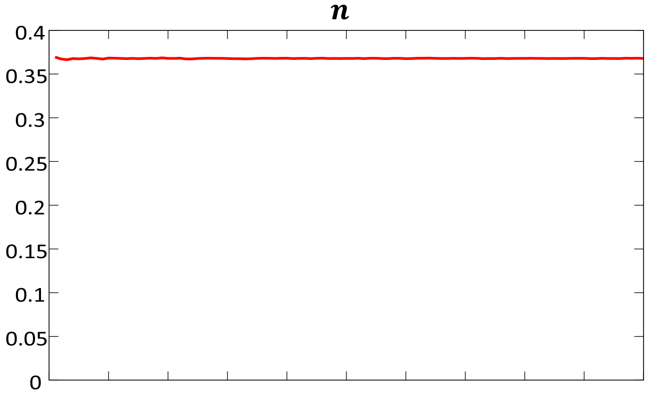
$$\frac{b(n)}{n}$$



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



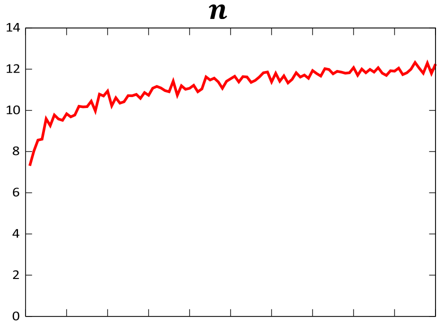
$$\frac{u(n)}{n}$$



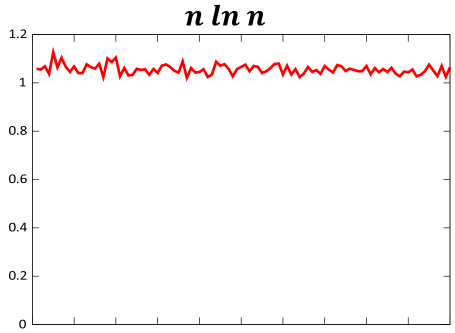
Powyższe wykresy zdecydowanie sugerują, że B_n jest złożoności $\Theta(\sqrt{n})$, czyli rośnie bardzo wolno, co potwierdza teorię **birthday paradox**.

Wykres z boku natomiast potwierdza, że U_n faktycznie zależy liniowo od n. Wynika z niego, że średnio w każdej próbie po wrzuceniu n kul, około 37% urn będzie pustych.

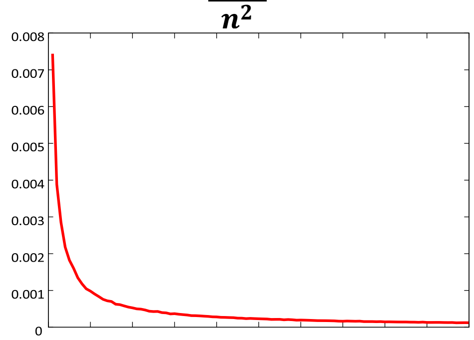
$$\frac{c(n)}{n}$$



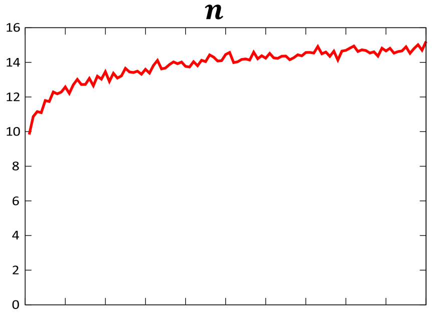
$$\frac{c(n)}{n \ln n}$$



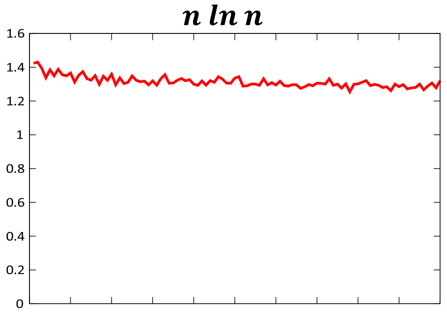
$$\frac{c(n)}{n^2}$$



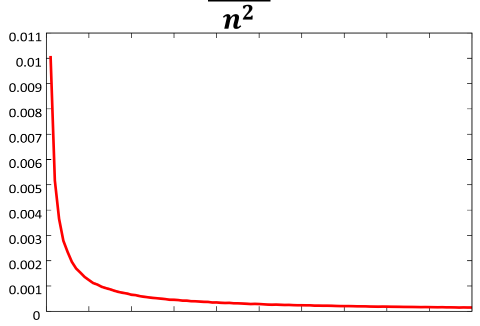
$$\frac{d(n)}{n}$$



$$\frac{d(n)}{n \ln n}$$

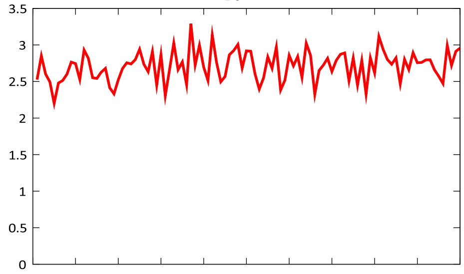


$$\frac{d(n)}{n^2}$$

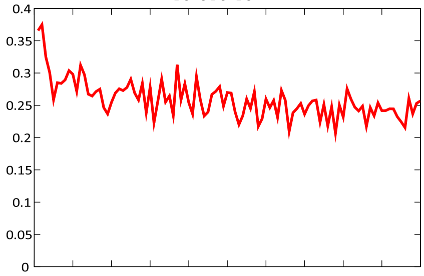


Zarówno C_n jak i D_n wyglądają na rosnące z prędkością n ln n. Dodatkowo $\frac{d(n)}{n \ln n}$ wygląda na powoli malejące i zbiegające do jakiejś stałej, co potwierdzałoby, że różni się od C_n o pewien niższy rząd asymptotyczny, ponieważ znaczenie tej różnicy maleje wraz ze wzrostem n.

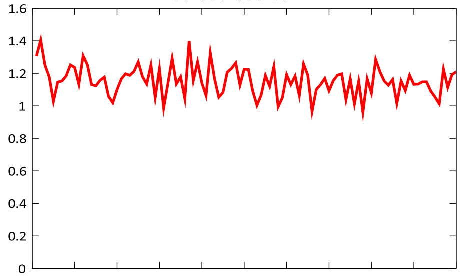
$$\boldsymbol{\frac{d(n) - c(n)}{n}}$$



$$\boldsymbol{\frac{d(n) - c(n)}{n \ln n}}$$



$$\boldsymbol{\frac{d(n) - c(n)}{n \ln \ln n}}$$



Mimo słabej widoczności można stwierdzić, że najbliżej stałej jest ostatni wykres –

tzn. $D_n – C_n$ jest złożoności $\Theta(n \ln \ln n)$.

Znaczenie **birthday paradox** dla funkcji hashujących

Przy tworzeniu hashy, zależy nam na tym, aby prawdopodobieństwo kolizji było jak najmniejsze. Birthday paradox mówi natomiast o tym, że to prawdopodobieństwo jest o wiele większe niż się może wydawać na pierwszy rzut oka, co może się okazać problematyczne przy implementacji funkcji hasujących.

Kod źródłowy: https://github.com/nskowron/Statistics2024/tree/main/Exc2