MPiS - Zadanie domowe 2

Michał Łukomski

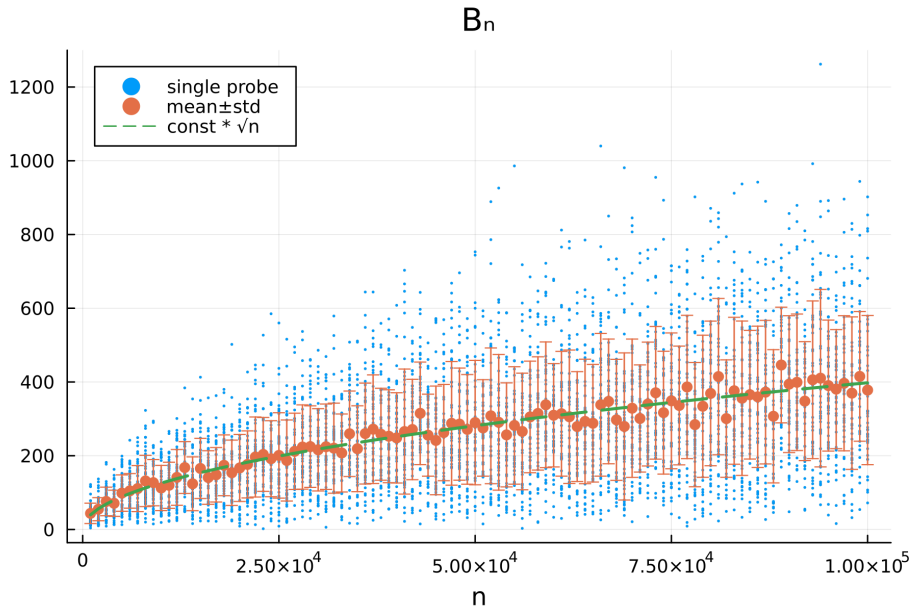
11 grudnia 2022

W symulacjach wrzucano kolejno kule do n urn. Każda kula wrzucana była niezależnie z jednakowym prawdopodobieństwem (równym $1/n$) do jednej z urn. Symulacje wykonano dla $n \in \{100, 2000, \ldots, 100000\}$, po $k = 50$ niezależnych powrótzeń dla każdego n. W trakcie symulacji zliczane były następujące statystyki:

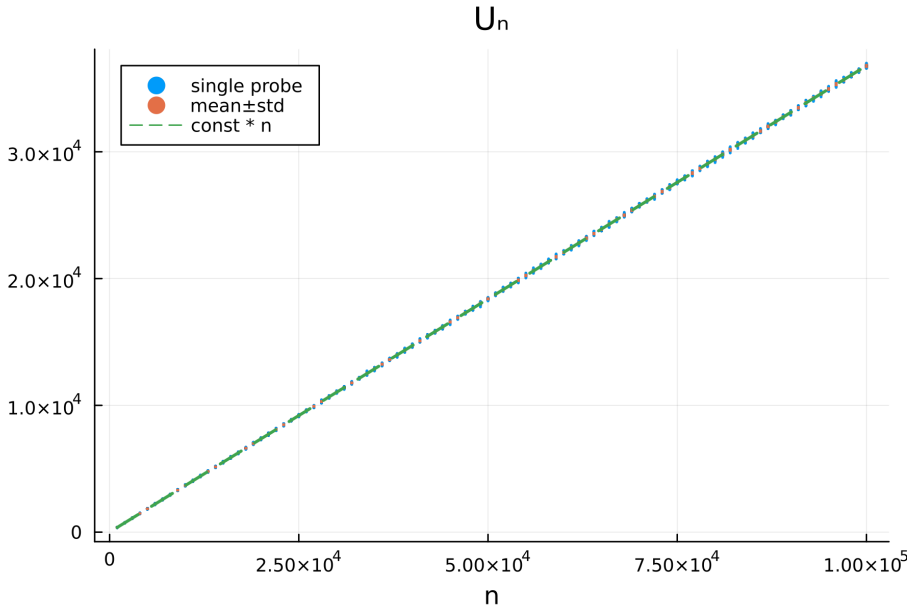
- B_n - moment pierwszej kolizji (*birthday paradox*)
- U_n - liczba pustych urn po wrzuceniu n kul
- L_n - maksymalna liczba kul w urnie po wrzuceniu n kul (*maximum load*)
- C_n - minimalna liczba rzutów, po której w każdej z urn jest co najmniej jedna kula (*coupon collector's problem*)
- D_n - minimalna liczba rzutów, po której w każdej z urn są co najmnije dwie kule (*siblings of the coupon collector*)
- $D_n - C_n$ - liczba rzutów od momentu C_n do momentu D_n

1 Wyniki

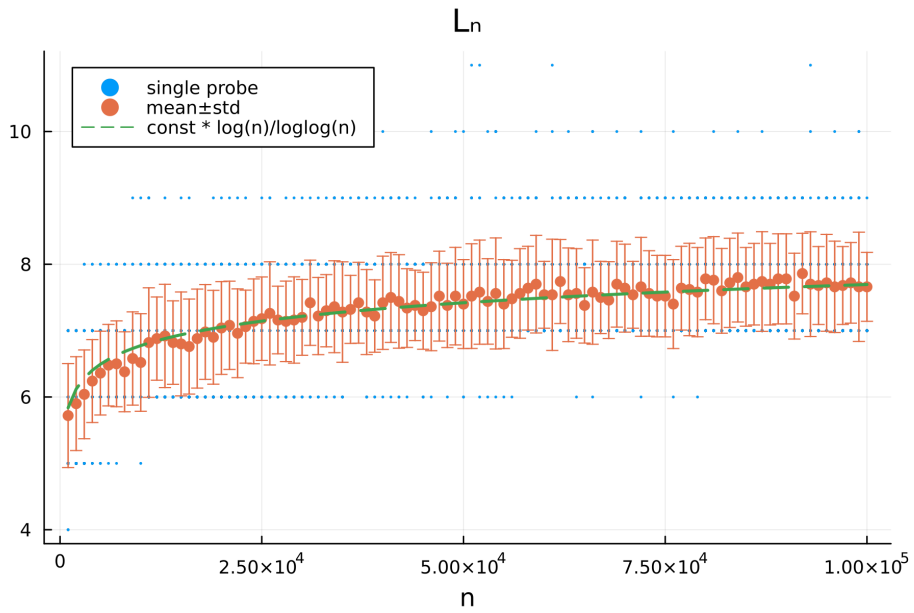
Wykresy przedstawiające otrzymane wyniki



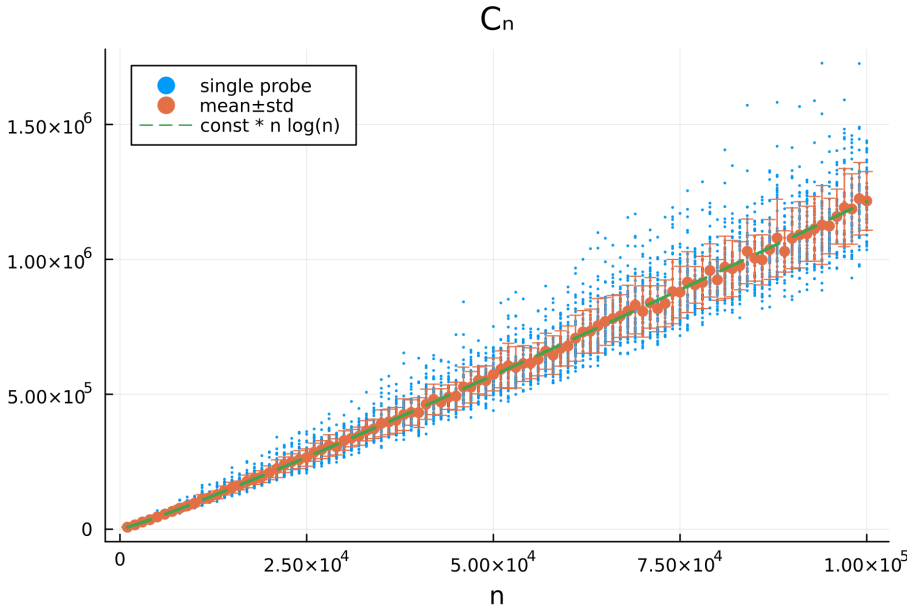
Rysunek 1: Moment pierwszej kolizji B_n, wraz z zaznaczonymi wartościami średnimi z k powtórzeń, oraz ich odchyleniem standardowym. Przy większej ilości urn (n) odchylenie standardowe, czyli rozrzut punktów zwiększa się, a ich średnia wartość rośnie proporcjonalnie do $const \cdot \sqrt{n}$. Stała $const$ została przyjęta jako średnia z wartości $\frac{B_n}{\sqrt{n}}$, nie jest to dokładne dopasowanie funkcji, a jedynie zgrubne pokazanie generalnego trendu.



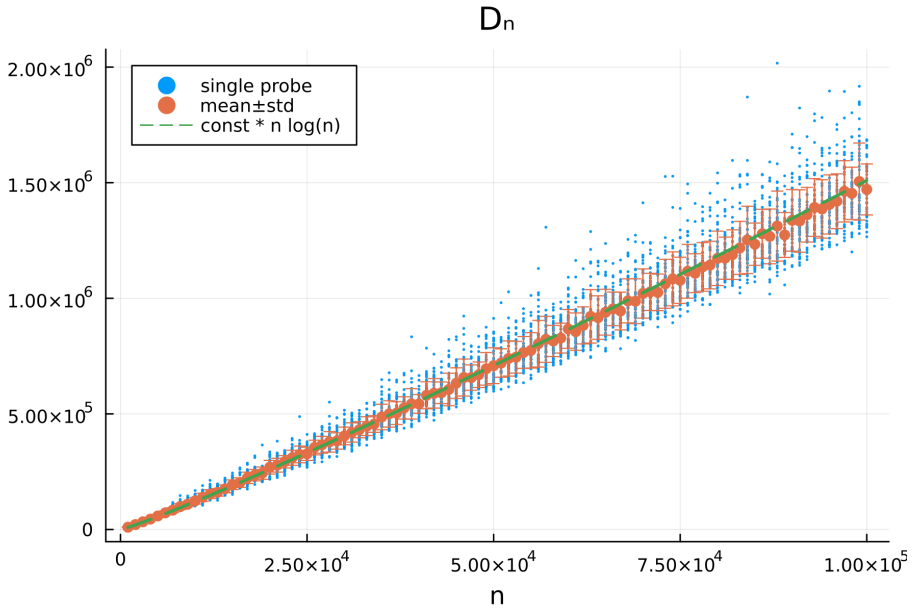
Rysunek 2: Liczba pustych urn U_n po wrzuceniu n kul, wraz z zaznaczonymi wartościami średnimi z k powtórzeń, oraz ich odchyleniem standardowym. Punkty są mocno skoncentrowane na prostej, na wykresie prawie nie widać ich rozrzutu, jest on w skali wykresu bardzo mały oraz niezależny od n. Wartości U_n rosną proporcjonalnie do $const \cdot n$. Stała $const$ została przyjęta jako średnia z wartości $\frac{U_n}{n}$, nie jest to dokładne dopasowanie funkcji, a jedynie zgrubne pokazanie generalnego trendu.



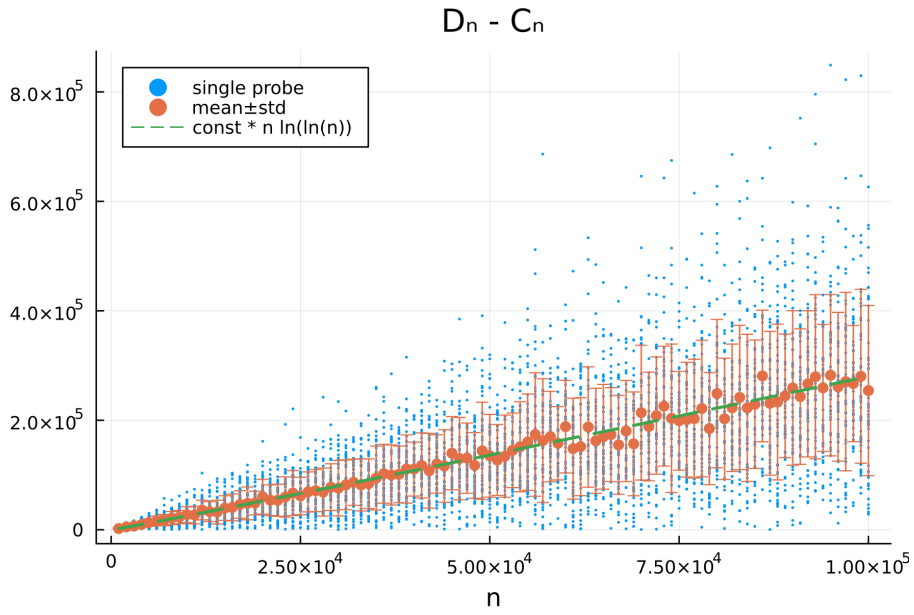
Rysunek 3: Maksymalna liczba kul w urnie L_n po wrzuceniu n kul, wraz z zaznaczonymi wartościami średnimi z k powtórzeń, oraz ich odchyleniem standardowym. Odchylenie standardowe wydaje się być niezależnie od n. Średnie wartości L_n rosną proporcjonalnie do $const \cdot \frac{\ln n}{\ln \ln n}$. Stała *const* została przyjęta jako średnia z wartości $\frac{L_n}{\frac{\ln n}{\ln \ln n}}$, nie jest to dokładne dopasowanie funkcji, a jedynie zgrubne pokazanie generalnego trendu.



Rysunek 4: Minimalna liczba rzutów C_n, po której w każdej z urn jest co najmniej jedna kula (*coupon collector's problem*), wraz z zaznaczonymi wartościami średnimi z k powtórzeń, oraz ich odchyleniem standardowym. Odchylenie standardowe (odpowiadające rozrzutowi punktów od średniej) zwiększa się wraz ze wzrostem n. Wartości C_n rosną proporcjonalnie do $const \cdot n \ln n$. Stała *const* została przyjęta jako średnia z wartości $\frac{C_n}{n \ln n}$, nie jest to dokładne dopasowanie funkcji, a jedynie zgrubne pokazanie generalnego trendu.



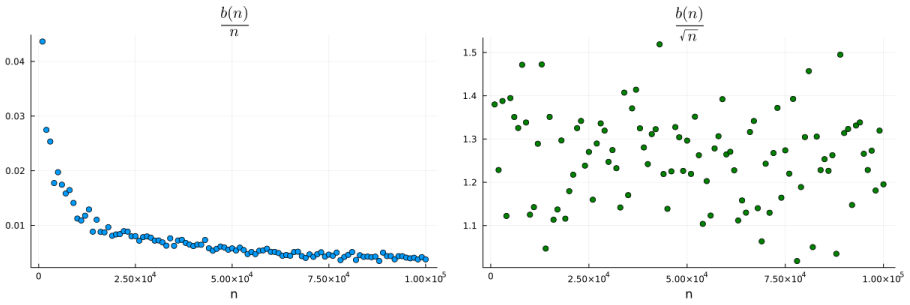
Rysunek 5: Minimalna liczba rzutów D_n, po której w każdej z urn są co najmniej dwie kule (*siblings of the coupon collector*), wraz z zaznaczonymi wartościami średnimi z k powtórzeń, oraz ich odchyleniem standardowym. Odchylenie standardowe (odpowiadające rozrzutowi punktów od średniej) zwiększa się wraz ze wzrostem n. Wartości D_n rosną proporcjonalnie do $const \cdot n \ln n$. Stała *const* została przyjęta jako średnia z wartości $\frac{D_n}{n \ln n}$, nie jest to dokładne dopasowanie funkcji, a jedynie zgrubne pokazanie generalnego trendu.



Rysunek 6: Liczba rzutów od momentu C_n (w każdej urnie jest przynajmniej jedna kula) do momentu D_n (w każdej urnie są przynajmniej dwie kule) wraz z icha wartościami średnimi oraz oddchyleniem standardowym. Rozrzut punktów od średniej (odchylenie standardowe) zwiększa się znacząco wraz ze wzrostem liczby urn. Wartości $D_n - C_n$ rosną proporcjonalnie do $const \cdot n \ln \ln n$. Stała $const$ została przyjęta jako średnia z wartości $\frac{D_n - C_n}{n \ln \ln n}$, nie jest to dokładne dopasowanie funkcji, a jedynie zgrubne pokazanie generalnego trendu.

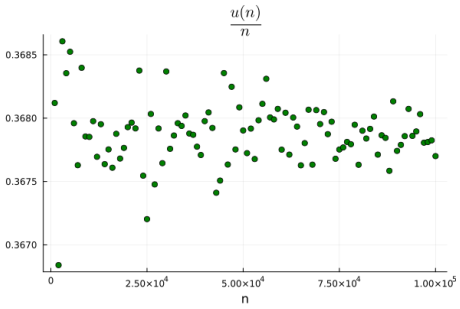
2 Pomocnicze wykresy

b, u, l, c, d oznaczają wartości średnie odpowiednio B_n, U_n, L_n, C_n, D_n. Narysowano dla nich wykresy, aby znaleźć funkcję do której są proporcjonalne.

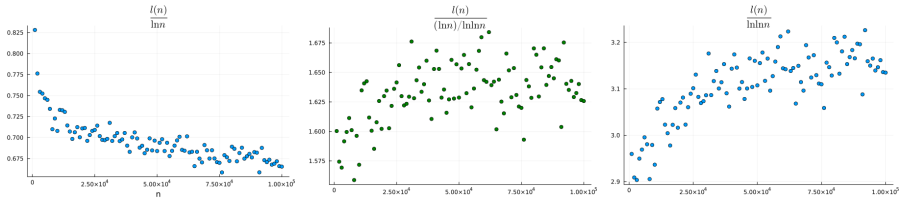


(a) $/n$ (b) $/\sqrt{n}$

Rysunek 7: Funkcje średniej wartości B_n od n, na zielono zaznaczono funkcję, która najbardziej wydaje się zbiegać do stałej (Rys. b). Dla dużych n średnia wartość B_n jest proporcjonalna do $\sqrt{n}$.

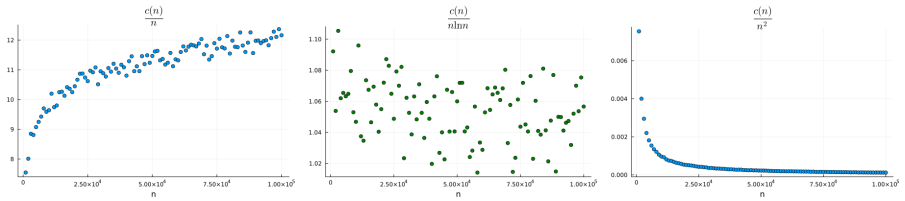


Rysunek 8: Funkcja średniej wartości U_n od n, wydaje się ona zbiegać do stałej, więc średnia wartość U_n zależy liniowo od n.



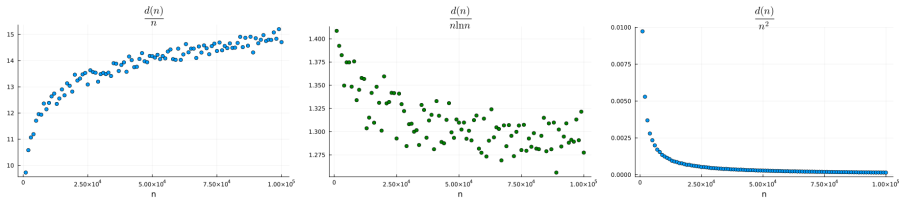
(a) $/\ln n$ (b) $/\frac{\ln n}{\ln\ln n}$ (c) $/\ln\ln n$

Rysunek 9: Funkcje średniej wartości L_n od n. Na zielono zaznaczono funkcję która wydaje się zbiegać do stałej (Rys. b). Dla dużych n średnia wartość L_n jest proporcjonalna do $\frac{\ln n}{\ln\ln n}$.



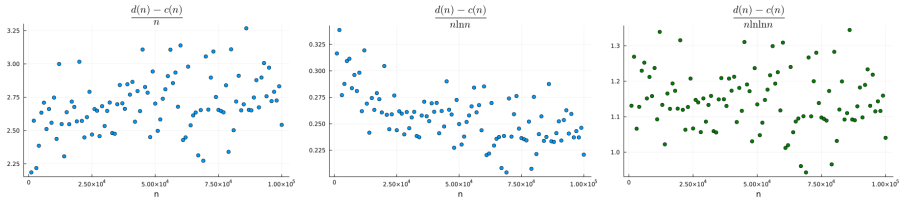
(a) $/n$ (b) $/n\ln n$ (c) $/n^2$

Rysunek 10: Funkcje średniej wartości C_n od n. Na zielono zaznaczono funkcję która wydaje się zbiegać do stałej (Rys. b). Dla dużych n średnia wartość C_n jest proporcjonalna do $n\ln n$.



(a) $/n$ (b) $/n\ln n$ (c) $/n^2$

Rysunek 11: Funkcje średniej wartości D_n od n. Na zielono zaznaczono funkcję która wydaje się zbiegać do stałej (Rys. b). Dla dużych n średnia wartość D_n jest proporcjonalna do $n\ln n$.



(a) $/n$ (b) $/n \ln n$ (c) $/n \ln \ln n$

Rysunek 12: Funkcje średniej wartości $D_n - C_n$ od n. Na zielono zaznaczono funkcję która wydaje się zbiegać do stałej (Rys. c). Dla dużych n średnia wartość C_n jest proporcjonalna do $n \ln \ln n$.

3 Wnioski

Do znajdywania asymptotyki wartości średnich korzystano z własności

$$x \sim f(n) \iff \frac{x}{f(n)} \sim const$$

Nie była to jednak do końca prawdziwa zależność, tylko jej przybliżenie, ponieważ w $f(n)$ uwzględniano tylko najbardziej liczący się człon, a pomijano te mniej znaczące. Jest jednak ono wystarczające na potrzeby tego eksperymentu.

Z pomocniczych wykresów wynika, że wartości średnie są zależne od n jak:

- $b(n) \sim \sqrt{n}$
- $u(n) \sim n$
- $l(n) \sim \frac{\ln n}{\ln \ln n}$
- $c(n) \sim n \ln n$
- $d(n) \sim n \ln n$
- $d(n) - c(n) \sim n \ln \ln n$

Minimalna liczba rzutów, po której w każdej z urn jest co najmniej jedna kula (C_n)stanowi analogię do *coupon collector's problem* - problemu kolekcjonera kuponów. W tym przypadku liczba różnych kuponów jest równa liczbie urn (n), a minimalna liczba rzutów C_n jest równa liczbie kuponów, które musi kolekcjoner zebrać, aby mieć jeden kupon każdego rodzaju. W problemie tym zakładamy, że każdy rodzaj kuponu ma takie samo prawdopodobieństwo wystąpienia, jak samo jak każda urna ma takie samo prawdopodobieństwo, że wpadnie do niej losowana kula. Liczba D_n natomiast odpowiada problemowi brata kolekcjonera kuponów (*coupon collector's brother*), w którym kolekcjoner musi zebrać wszystkie kupony dla siebie, oraz swojego brata, czyli efektywnie musi mieć po 2 kupony każdego rodzaju. W jeszcze ogólniejszej wersji tego problemu - *the siblings of the coupon collector* kolekcjoner musi zebrać po k kuponów każdego rodzaju, gdzie k jest liczbą jego rodzeństwa, to uogólnienie nie było jednak rozważane w tym eksperymencie.

Liczba pierwszej kolizji, czyli moment w krótym pierwszy raz kula wpadła do niepustej urny nazywa się *birthday paradox*, z uwagi na podobieństwo do paradoksu urodzinowego, zgodnie z którym w grupie 23 osób prawdopodobieństwo, że dwie osoby urodziły się tego samego dnia wynosi 50%. Liczba urn n jest analogiczna do liczby dni ($n = 365$ w paradoksie urodzinowym), a B_{365} jest analogiczne do liczby osób które spotykamy, dopóki dwie z tych osób nie urodziły się tego samego dnia w roku.

Birthday paradox w kontekście funkcji hashujących jest powiązany z częstością występowania kolizji w tych funkcjach. Jeśli n będzie liczbą wszystkich możliwych kombinacji hashy, które na wyjściu może wygenerować funkcja, to B_n nam powie ile przedmiotów możemy dodawać do zbioru, aż do momentu gdy

dwa z nich dostaną ten sam hash (z zastrzeżeniem, że w eksperymencie *balls and bins* każda kula była wrzucana niezależnie od innych, a urny były losowane z rozkładu jednorodnego, co nie zawsze jest spełnione w fukcjach hashujących, szczególnie tych niekryptograficznych - nie każdy hash ma taką samą "szansę" na bycie trafionym, oraz "odległość" między kluczami może mieć wpływ na wyjściowy hash). Stwarza to np problem w słownikach (*hashmap*), gdzie jeśli dodamy zbyt wiele elementów i wystąpi kolizja, to stracimy dane. Wybiera się więc takie funkcje hashujące, które zwracają odpowiednio dużą ilość możliwych hashy, aby kolizja nie nastąpiła w praktycznych zastosowaniach, mając też na uwadze szybkość działania funkcji.

Kryptograficzne funkcje hashujące to taka podgrupa funkcji hashujących, w których mniej liczy się szybkość wykonywania funkcji, a bardziej jej niezawodność i odporność na ataki. Czyli dopuszczalny moment pierwszej kolizji (B_n) będzie dużo późniejszy (oczekiwane B_n będzie dużo większe). Kryptograficzne funkcje hashujące powinny też być odporne na ataki, w których ktoś będzie chciał doprowadzić do kolizji, więc np. "podobieństwo" elementów wejściowych nie powinno mieć znaczenia, tak jak przy losowaniu urny nie mają znaczenia kule wrzucone wcześniej, ponieważ każde zdarzenie jest niezależne od poprzednich.

Eksperyment wykonano w języku Julia 1.7.2, jako generator liczb losowych wykorzystano Xoshiro256++ - domyślnego generatora języka Julia.