Wrocław, 02.01.2022

Metody Probabilistyczne i Statystyka

Lista 7

- część eksperymentalna

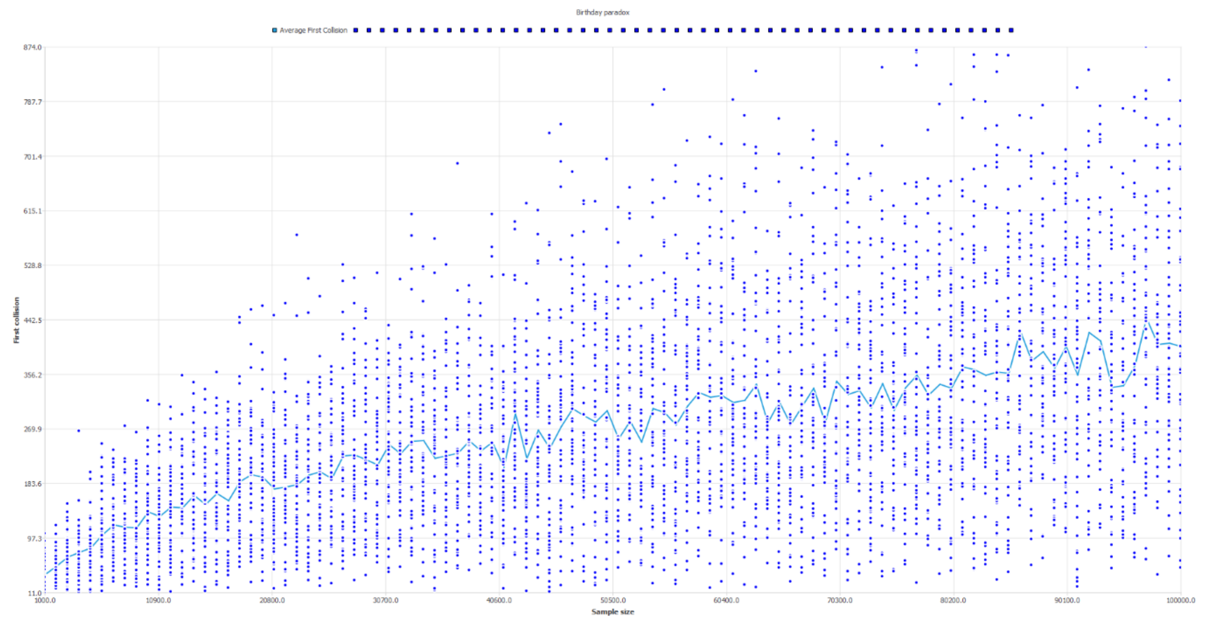
Małgorzata Orłowska

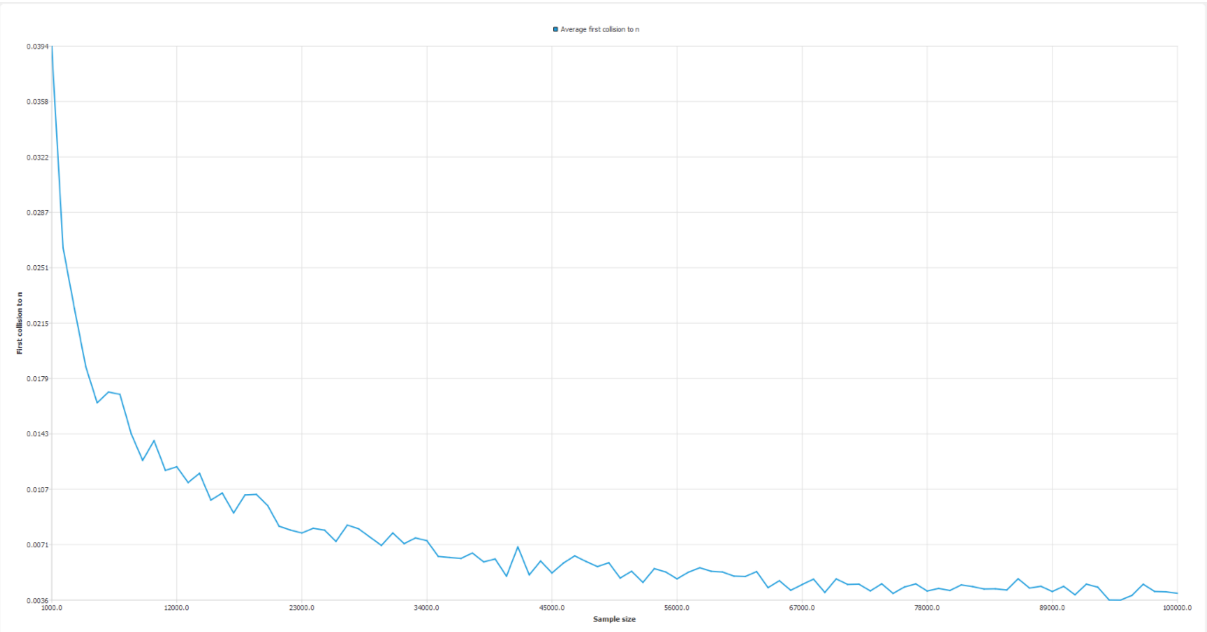
Poniższe eksperymenty wykonano przy pomocy generatora Mersenne Twister, a wykresy wygenerowano używając narzędzia Qt.

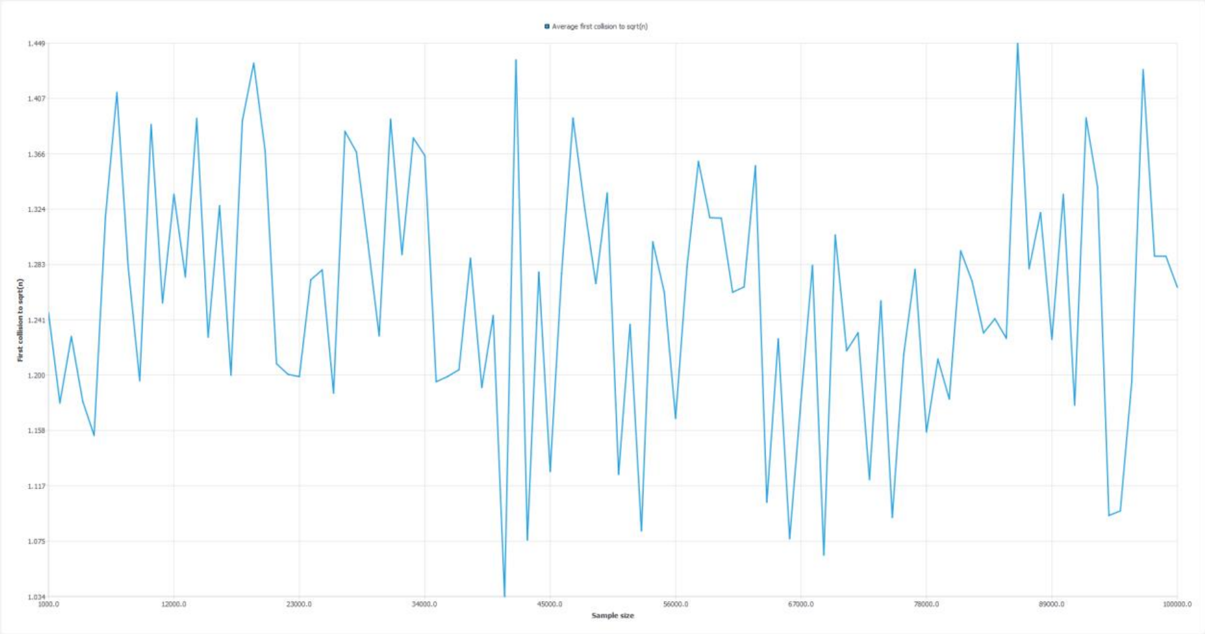
Liczba powtórzeń dla wszystkich symulacji k = 50.

Zad. 1

1. Moment pierwszej kolizji, pierwsza kula, która trafiła do niepustej urny (birthday paradox).

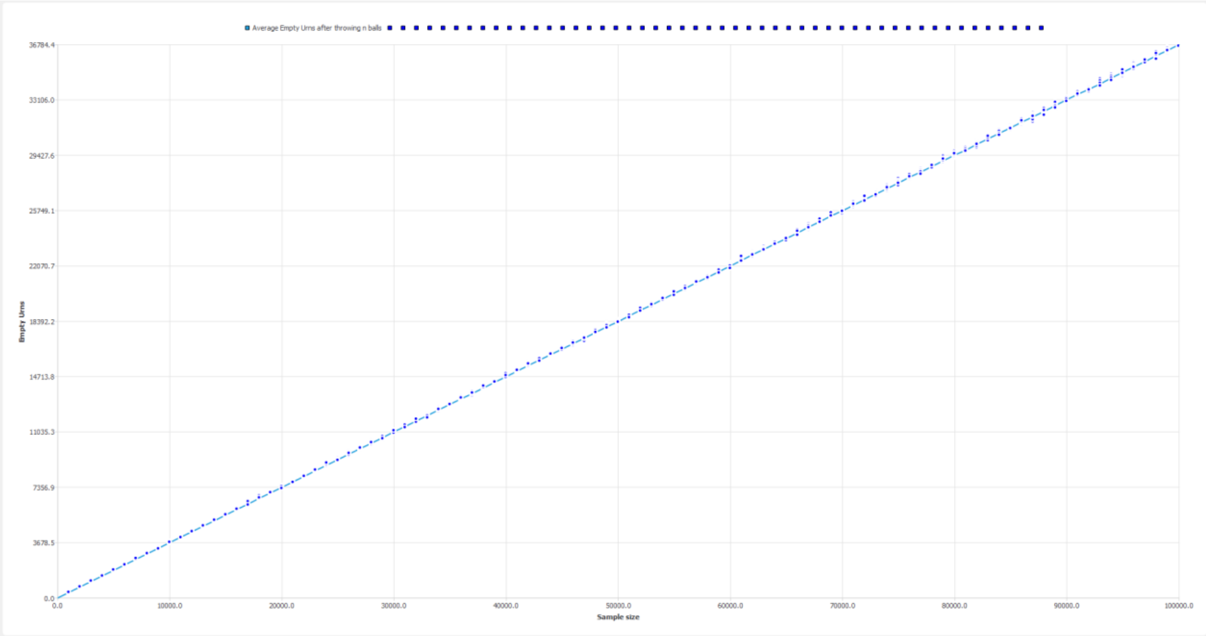


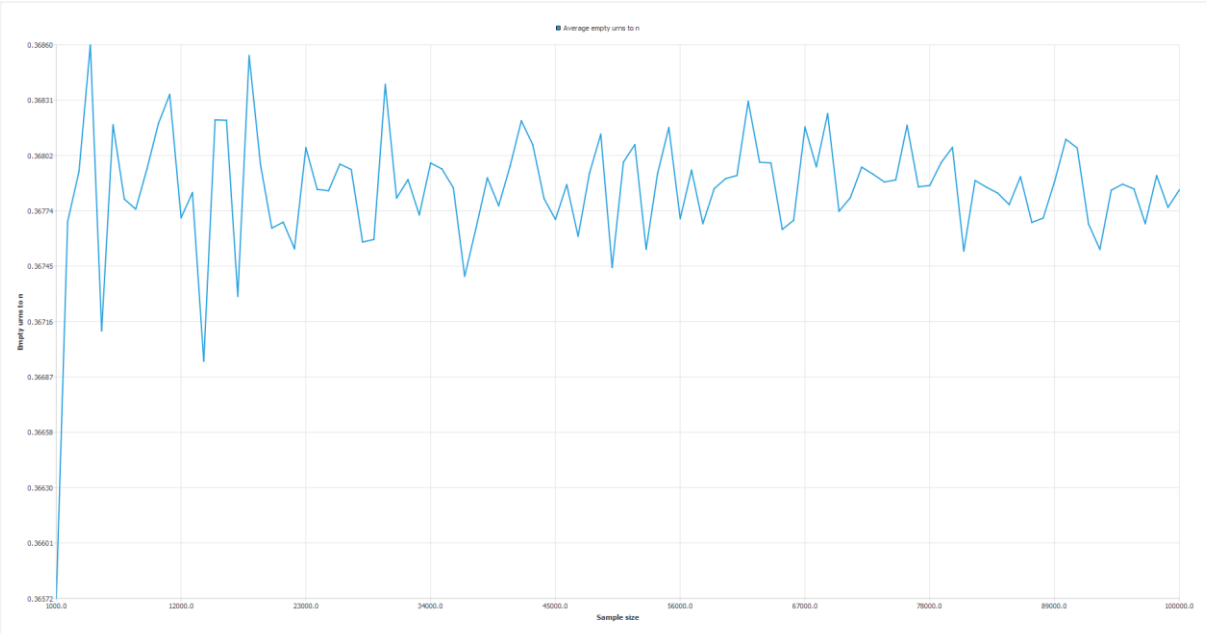




Wniosek: Średni moment pierwszej kolizji oscyluje wokół 1.28*sqrt(liczba urn), jest zatem „szybki”. Zgadza się to z tzw. paradoksem urodzin – eksperymentalnie wyznaczona stała sugeruje że przy 365-dniowym roku osobowej klasie mamy moment pierwszej kolizji ~24,5, czyli można oczekiwać że w 25-osobowej klasie znajdą się co najmniej dwie osoby z urodzinami tego samego dnia.

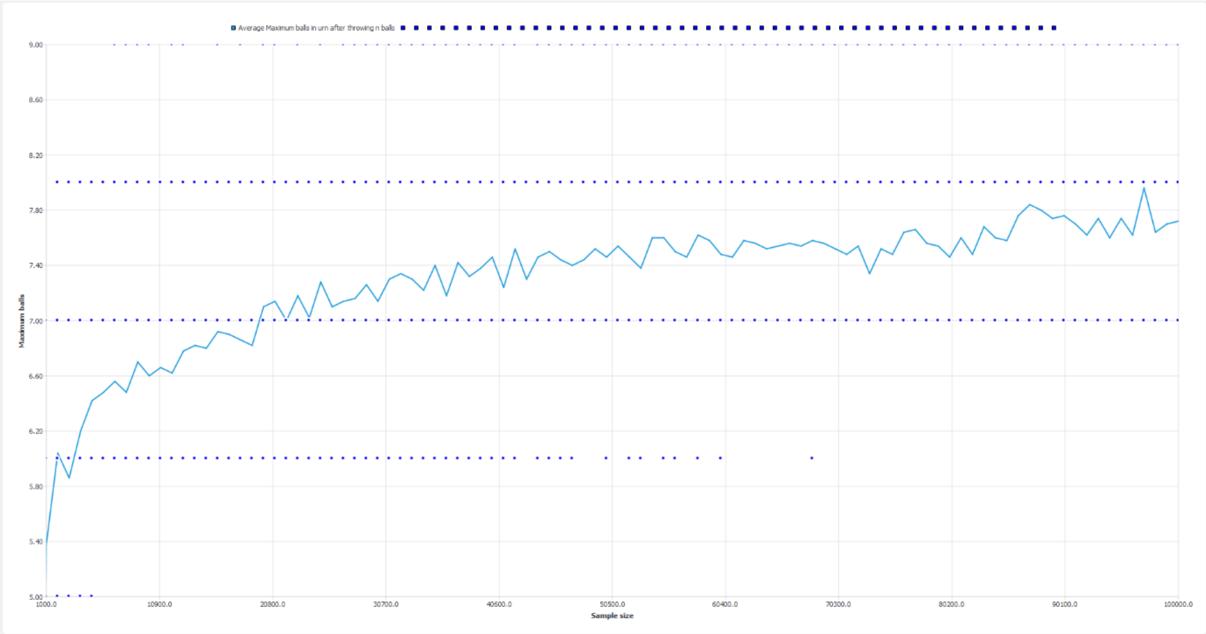
2. Liczba pustych urn po wrzuceniu n kul.

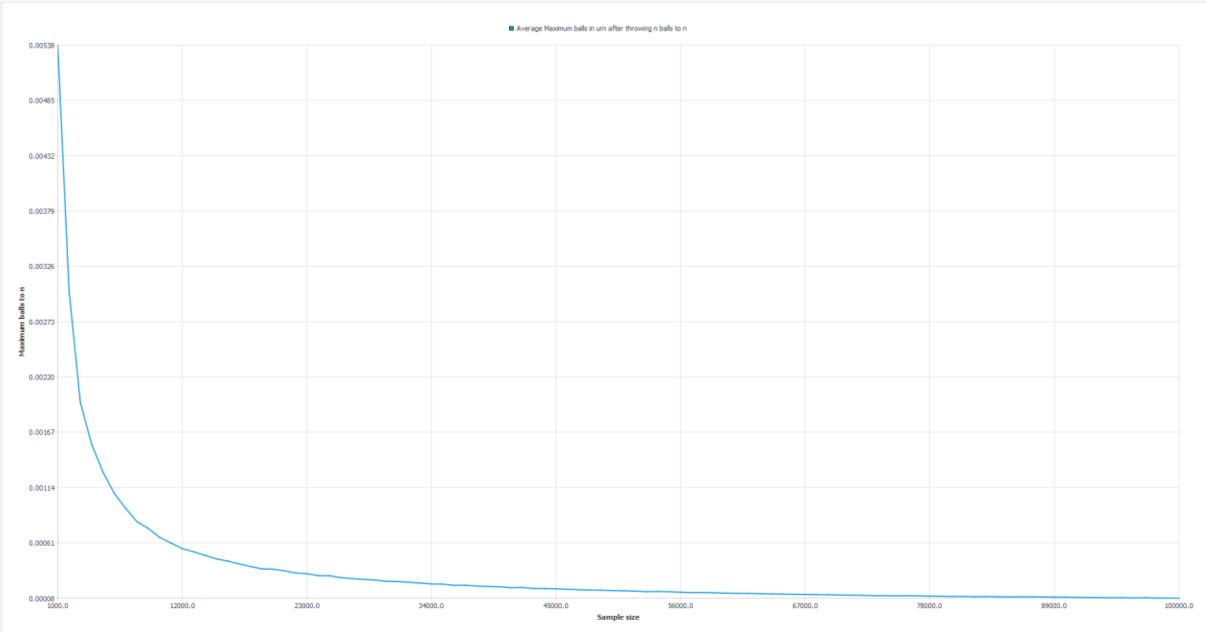


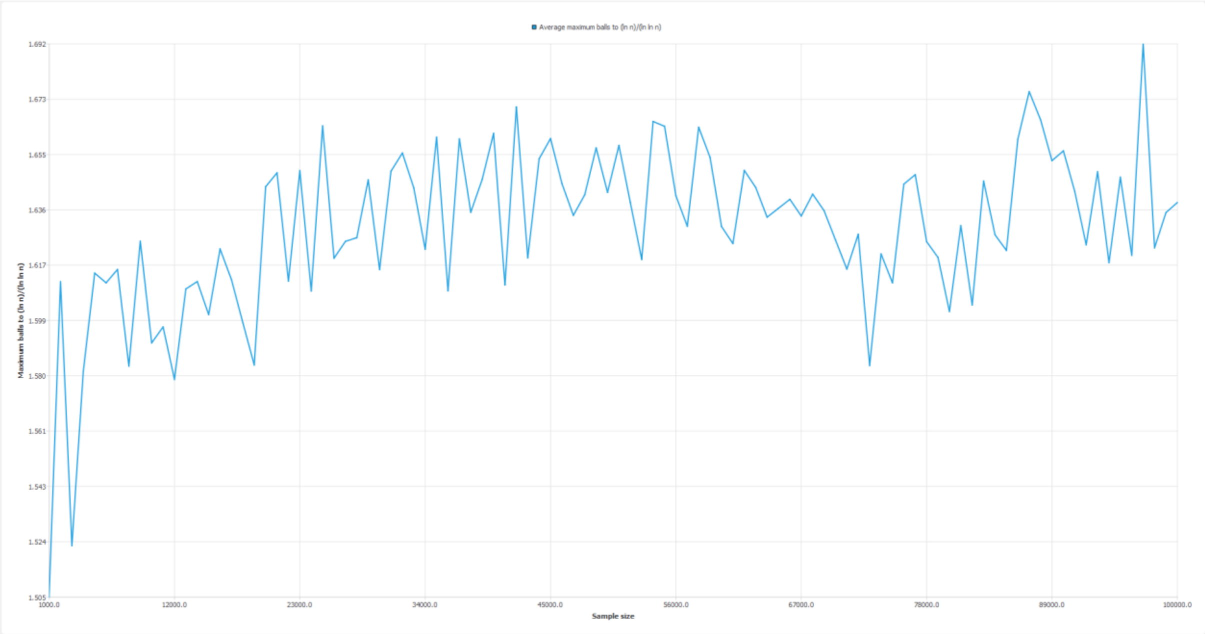


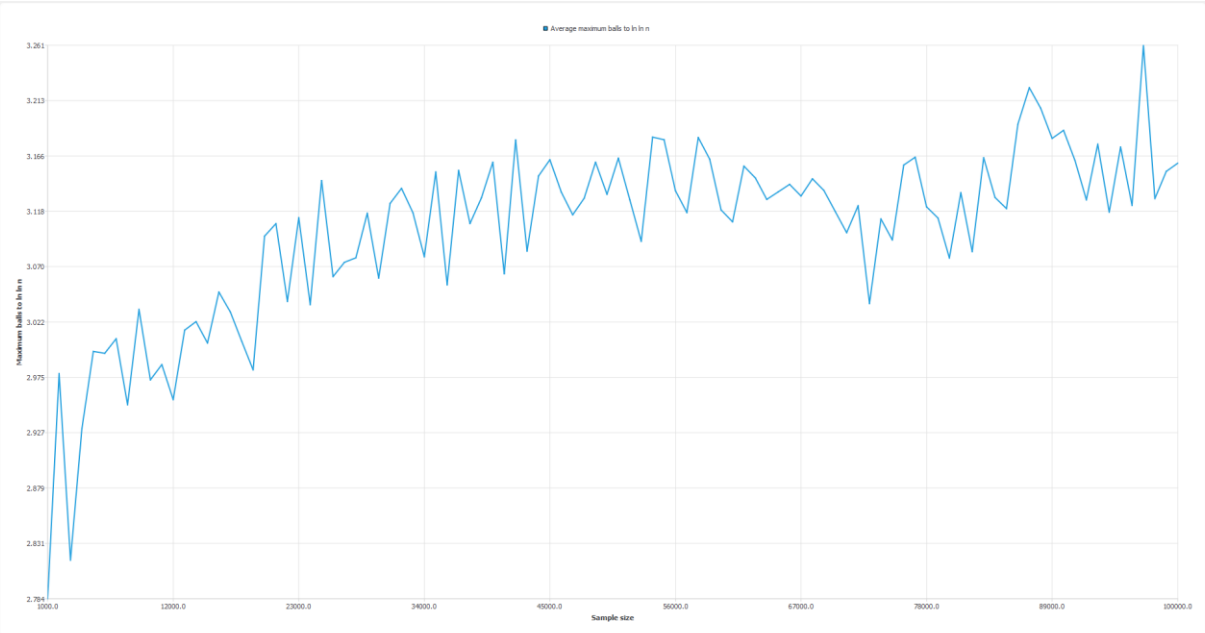
Wniosek: Średnia liczba pustych urn po wrzuceniu n kul oscyluje wokół 0.367*n.

3. Maksymalna liczba kul w urnie po wrzuceniu n kul.



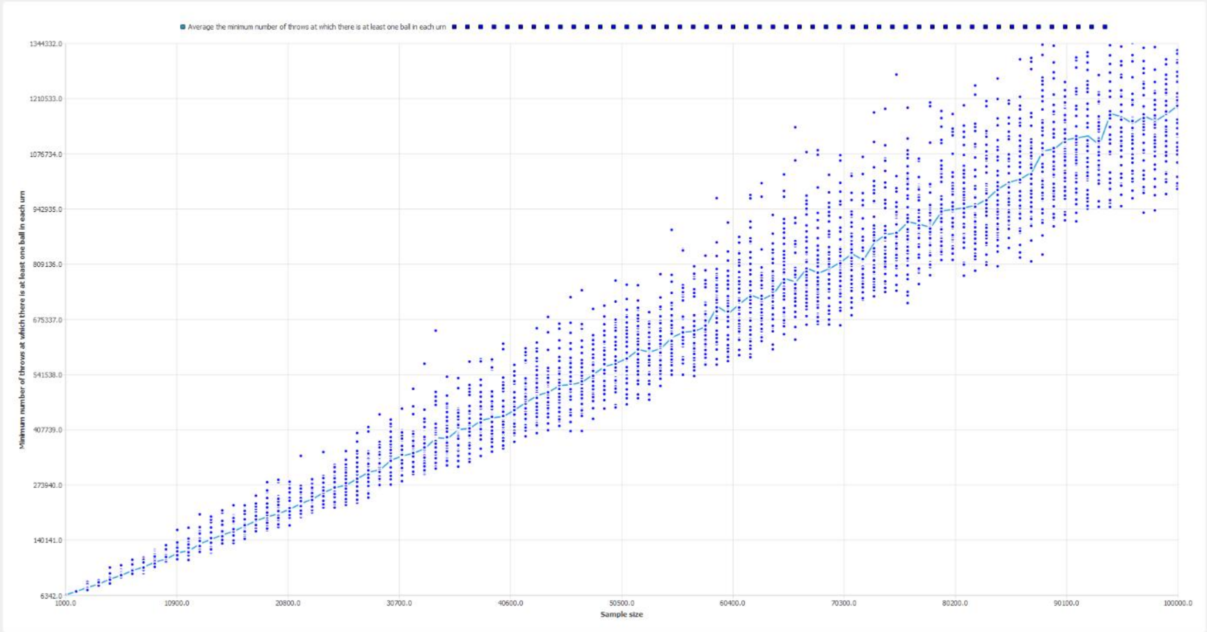


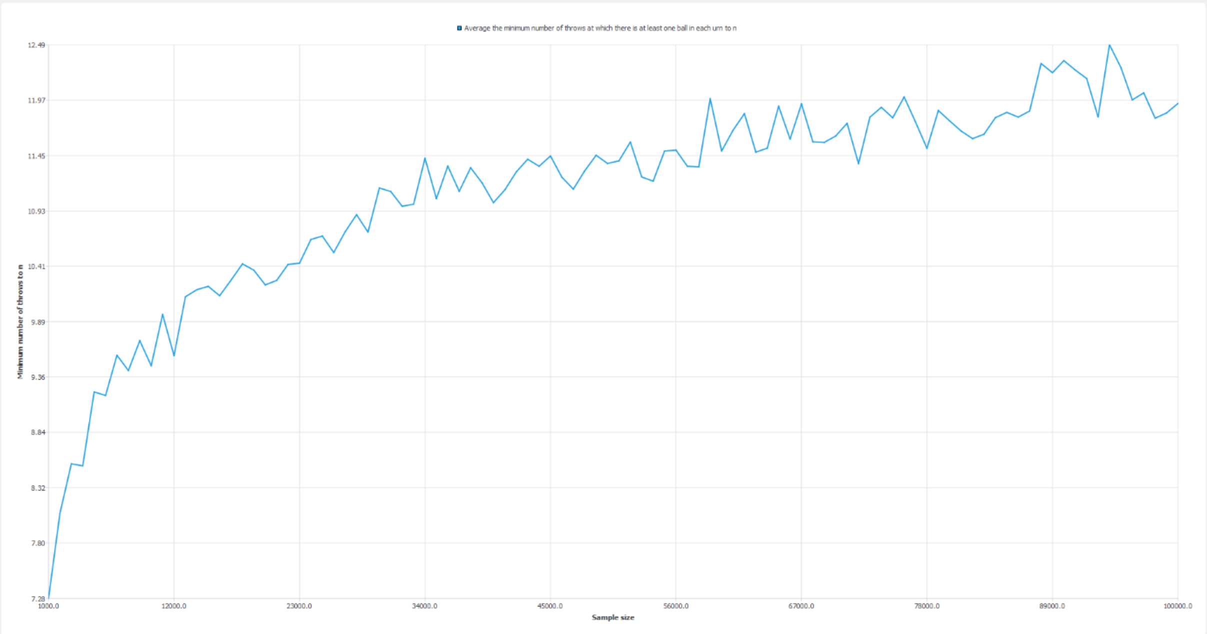


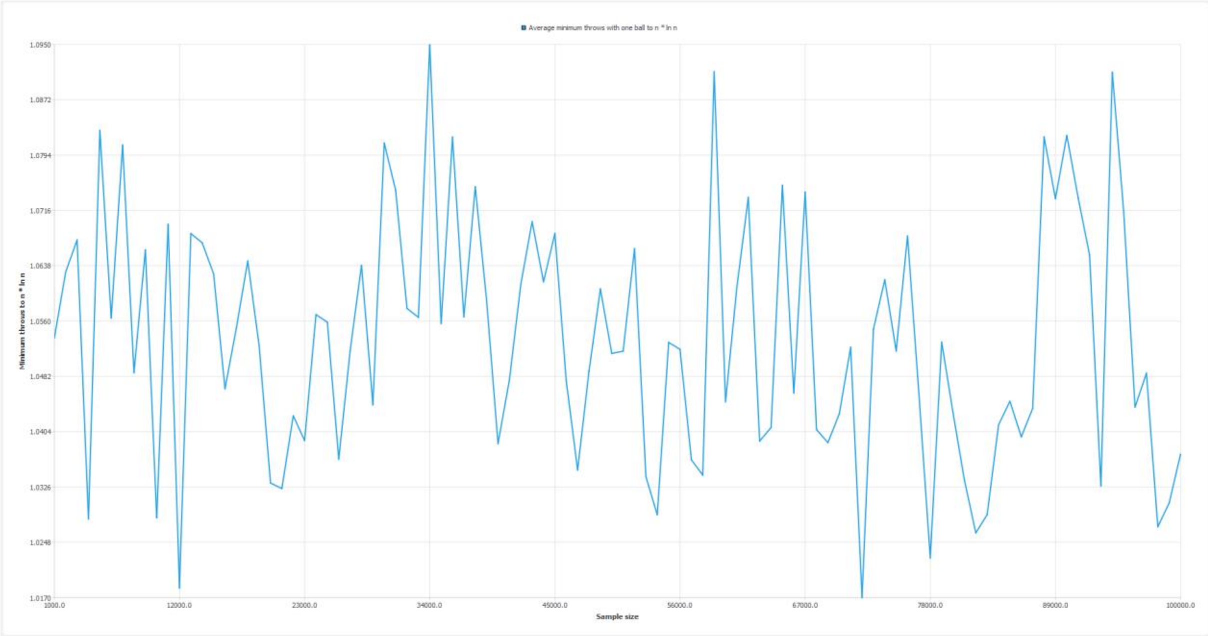


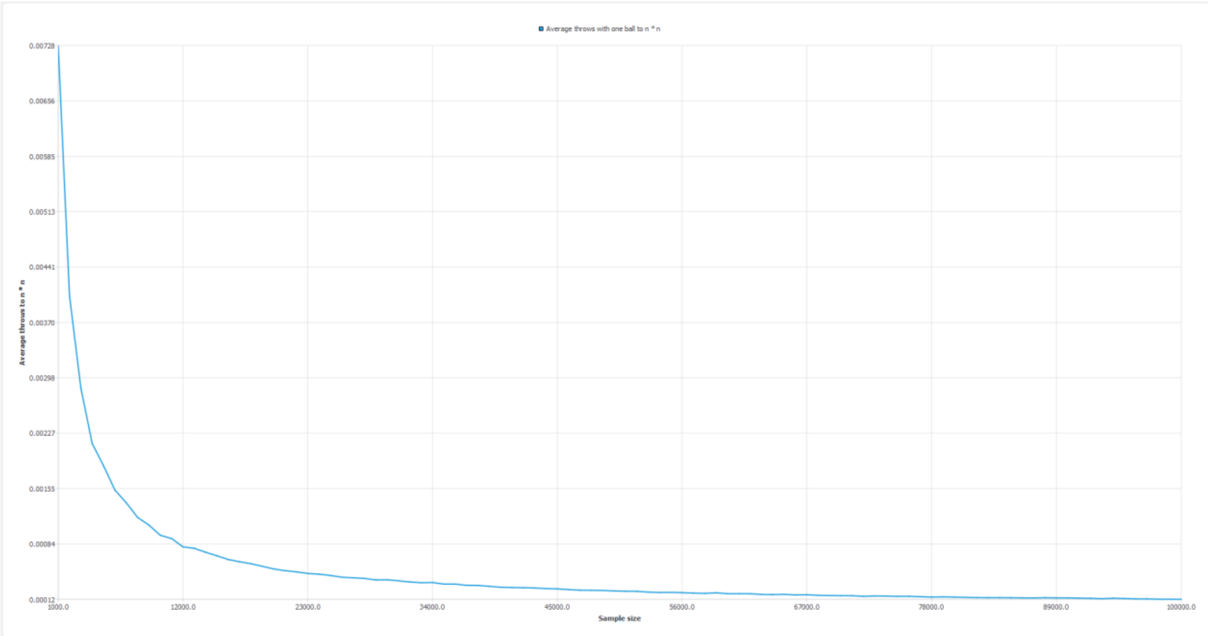
Wniosek: Eksperymentalnie w żadnej z prób nie udało się osiągnąć więcej niż 9 kul. Średnia maksymalna liczba kul w urnie po wrzuceniu n kul oscyluje wokół 1.6*((ln(n))/(ln(ln(n))).

4. Minimalna liczba rzutów, po której w każdej z urn jest co najmniej jedna kula (problem kolekcjonera kuponów).



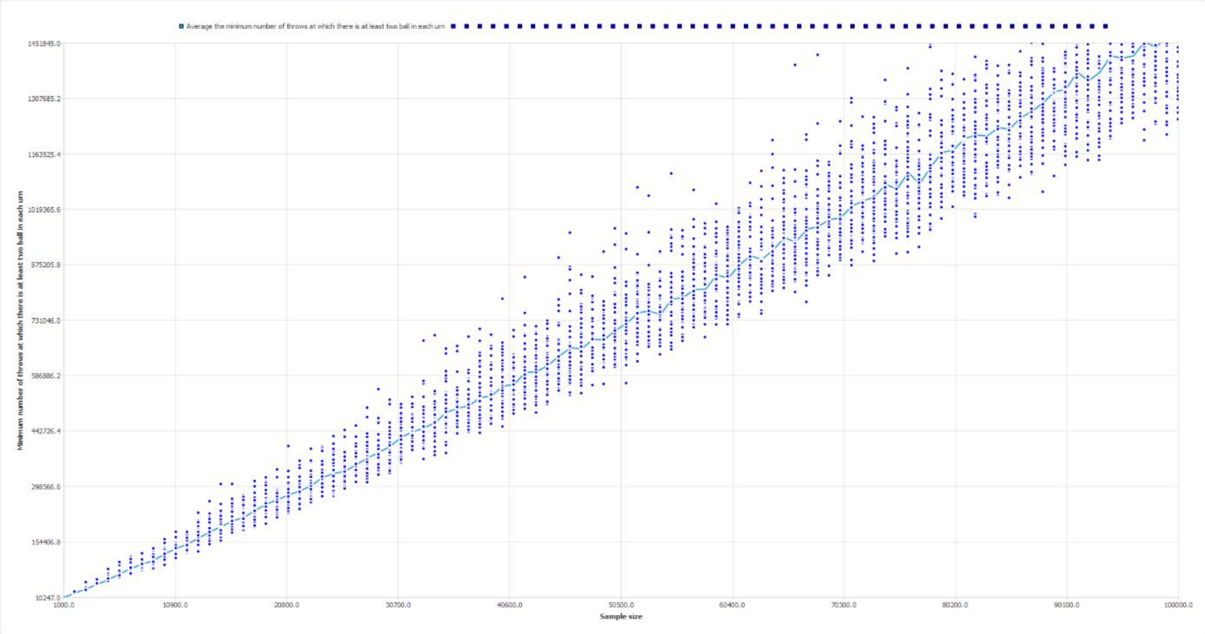


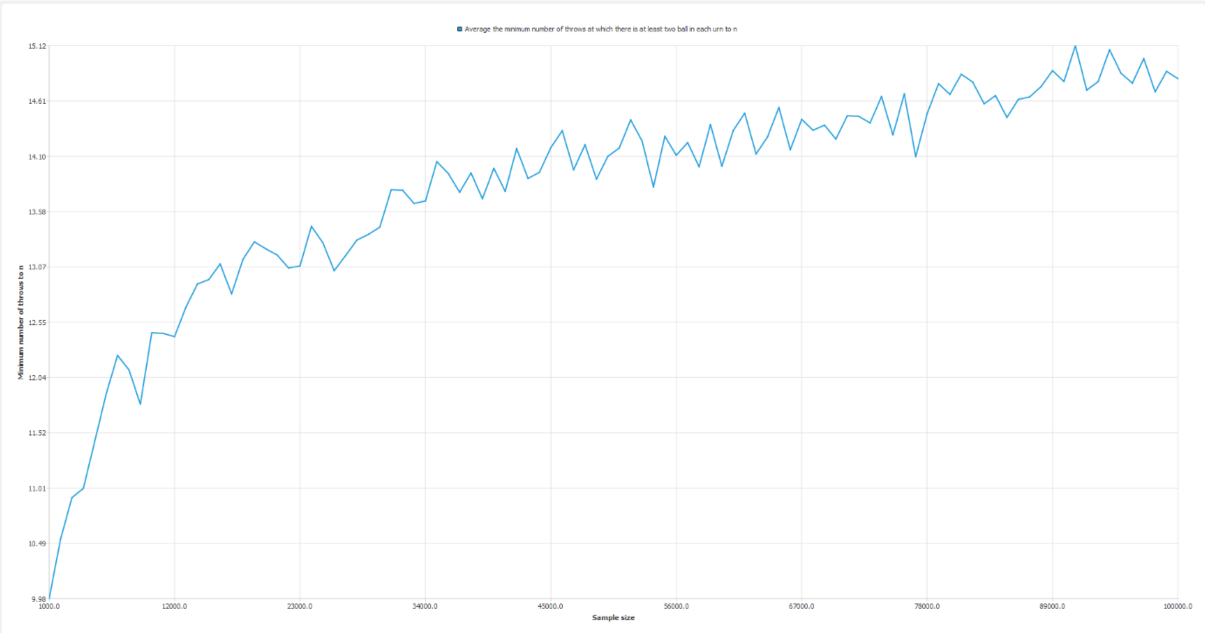


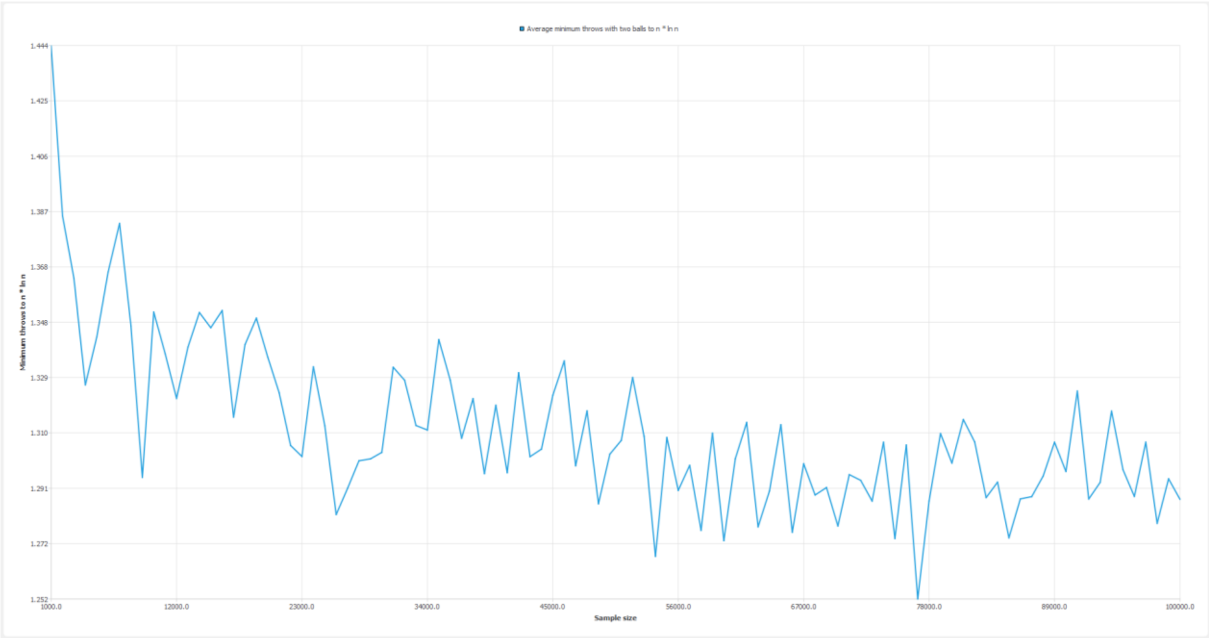


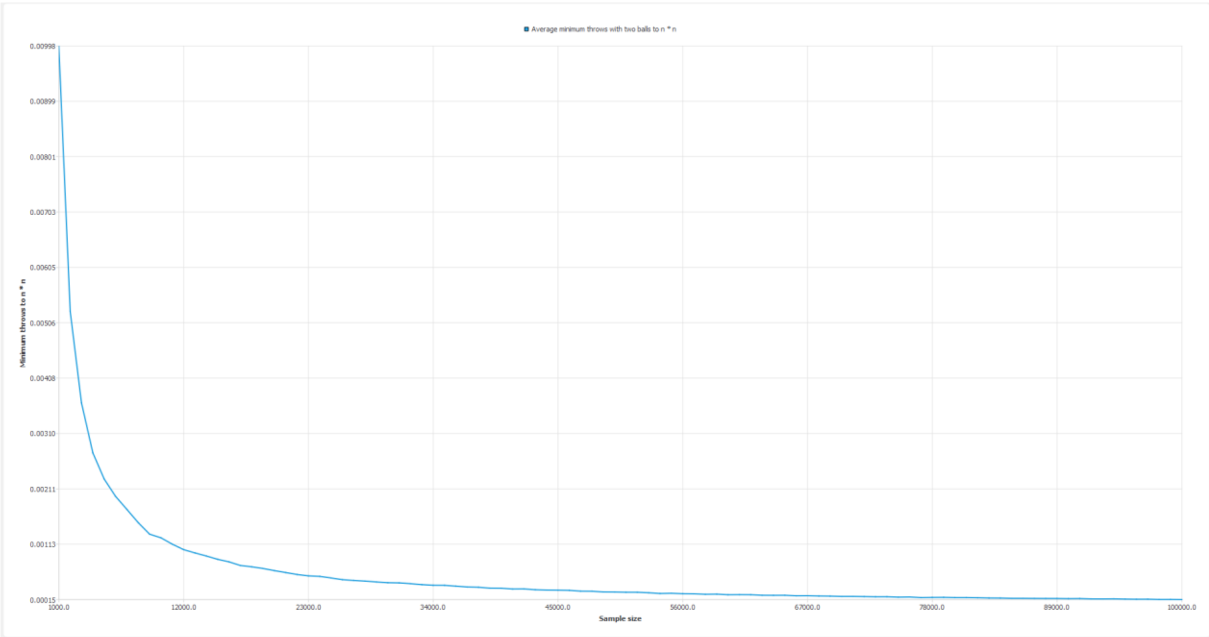
Wniosek: Uzyskane wyniki koncentrują się wokół wartości średniej. Średnia minimalna liczba rzutów, po której w każdej z urn jest co najmniej jedna kula oscyluje wokół 1.05*n*ln(n).

5. Minimalna liczba rzutów, po której w każdej z urn są co najmniej dwie kule.



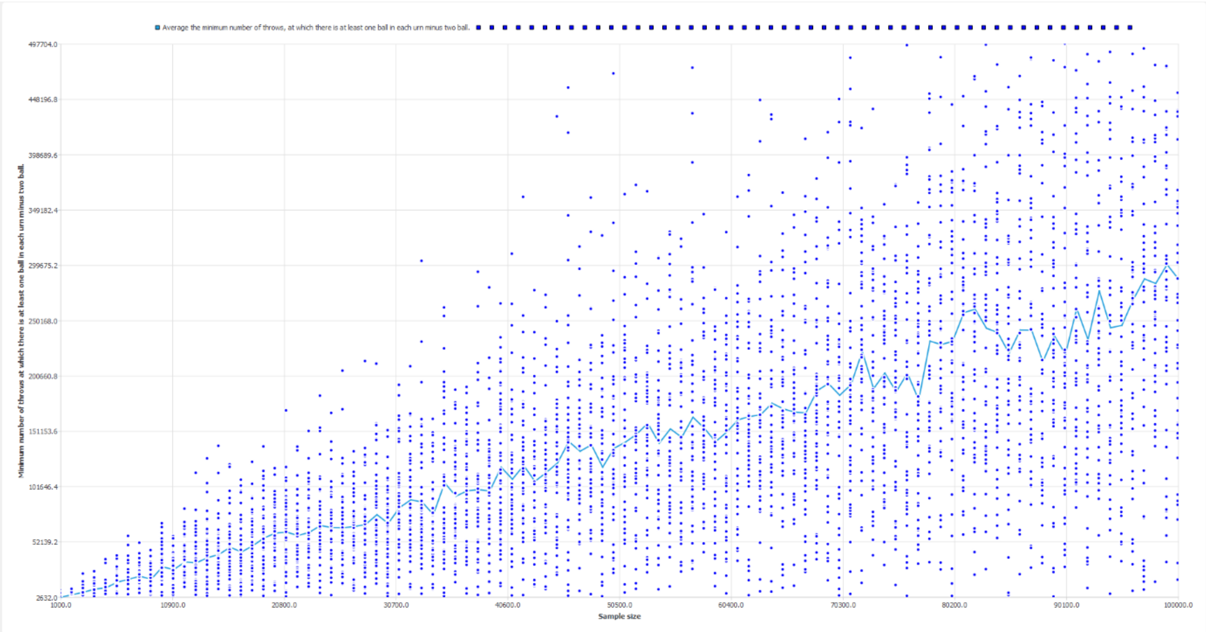


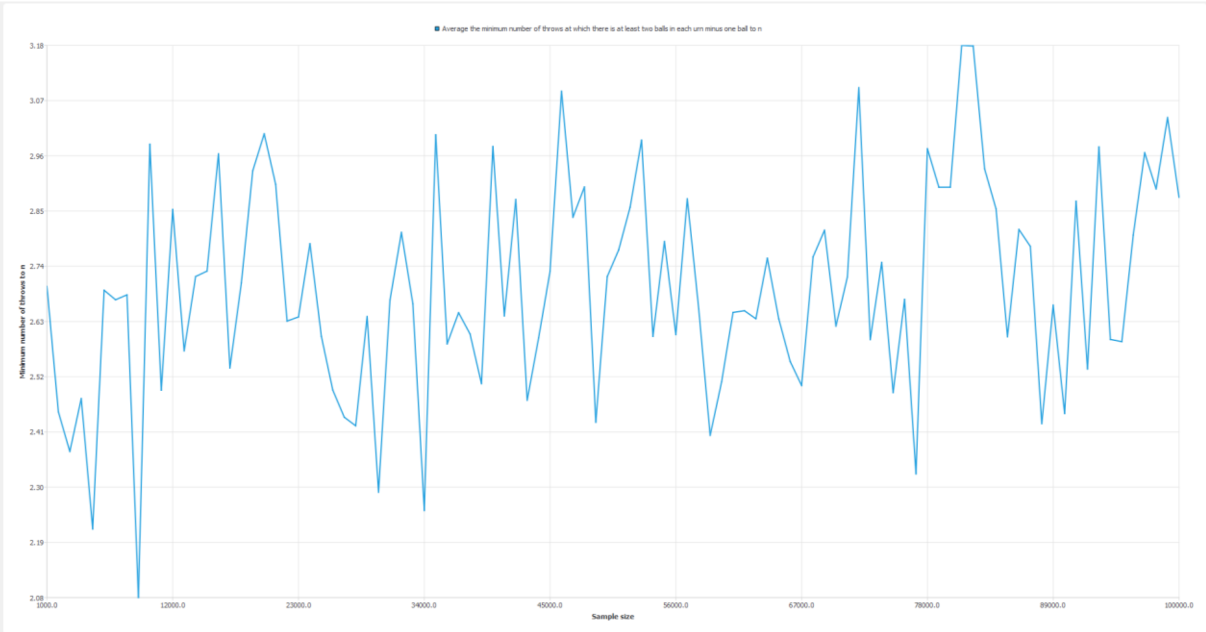


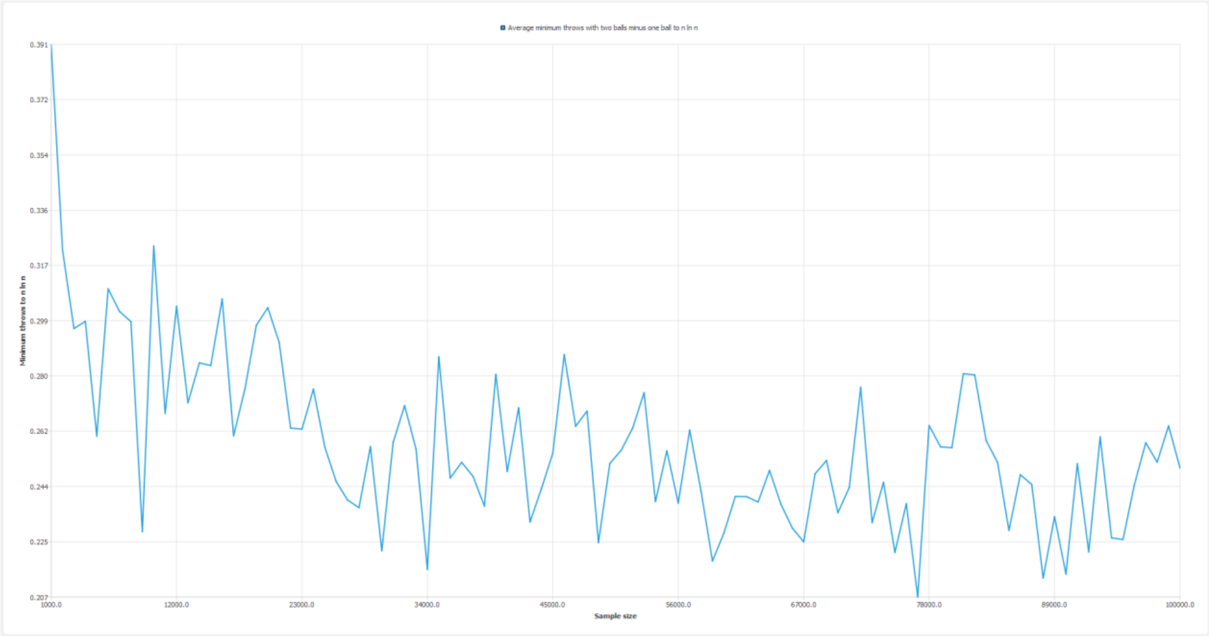


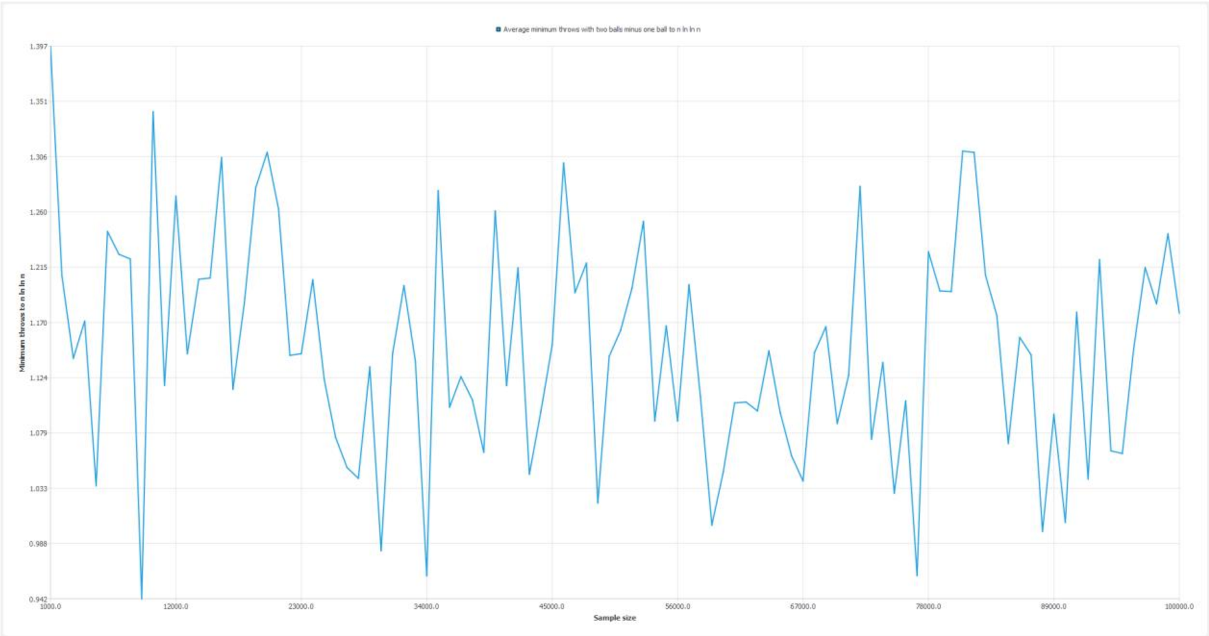
Wniosek: Uzyskane wyniki koncentrują się wokół wartości średniej. Średnia minimalna liczba rzutów, po której w każdej z urn są co najmniej dwie kul oscyluje wokół 1.3*n*ln(n).

6. Różnica pomiędzy minimalną liczbą rzutów, po której w każdej z urn są co najmniej dwie kule a minimalną liczbą rzutów, po której w każdej z urn jest co najmniej jedna kula.









Wniosek: Średnia różnica pomiędzy minimalną liczbą rzutów, po której w każdej z urn są co najmniej dwie kule a minimalną liczbą rzutów, po której w każdej z urn jest co najmniej jedna kula oscyluje wokół 1.17*n*ln(ln(n))

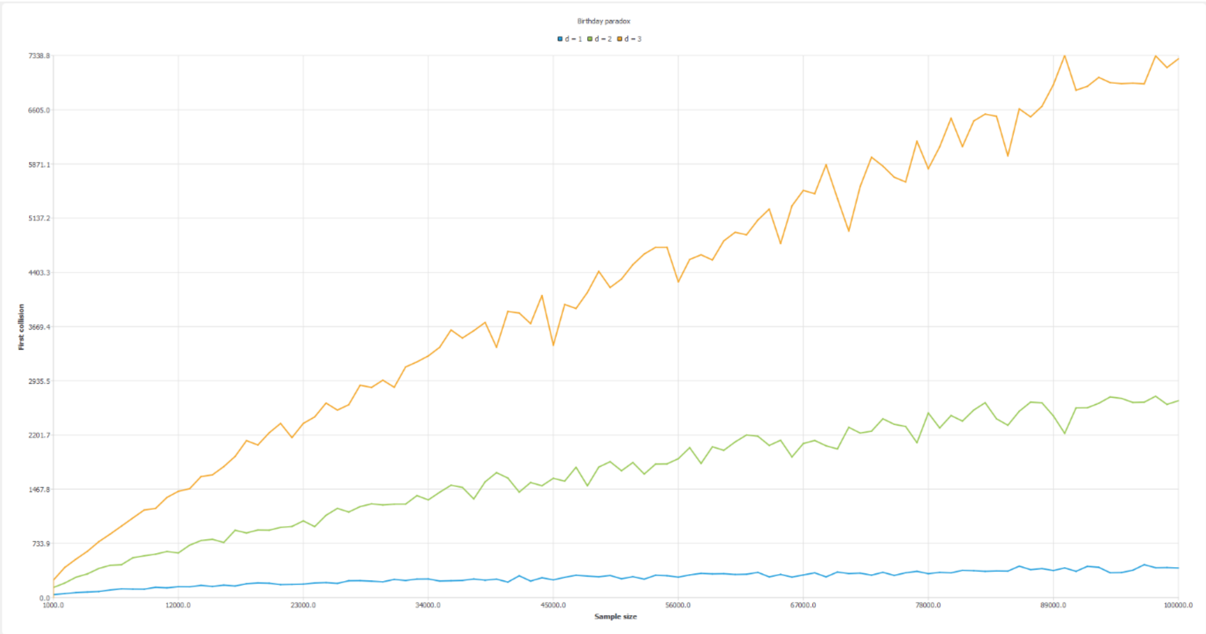
Zad. 2

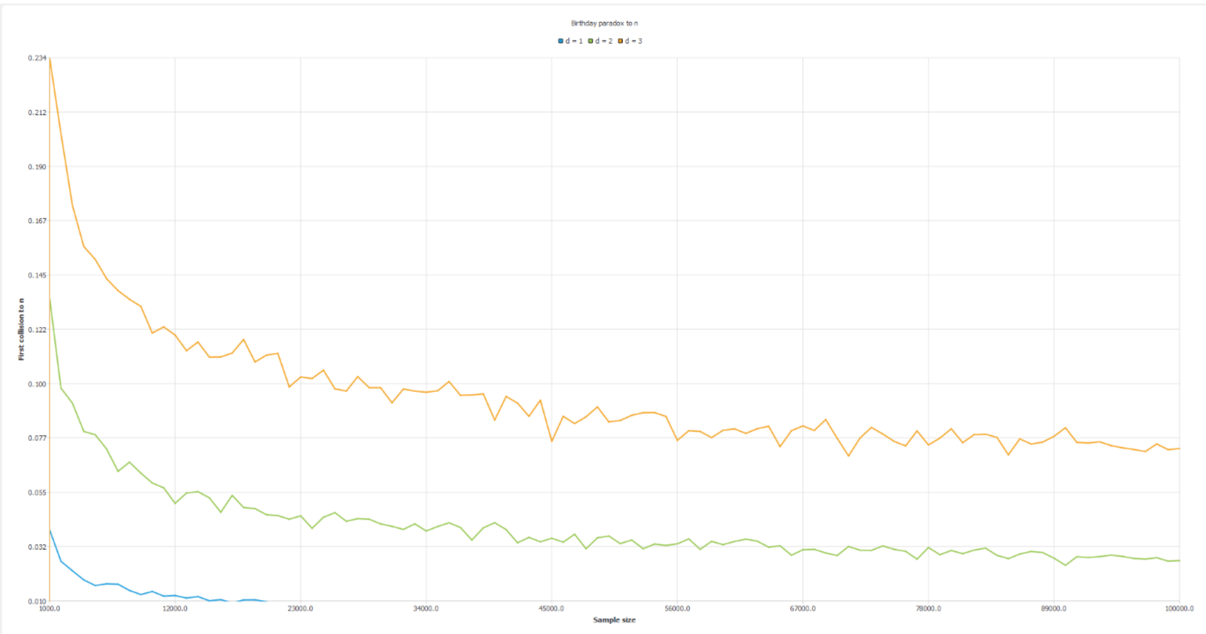
Eksperyment jak w zadaniu 1, ale zmodyfikowany w następujący sposób.

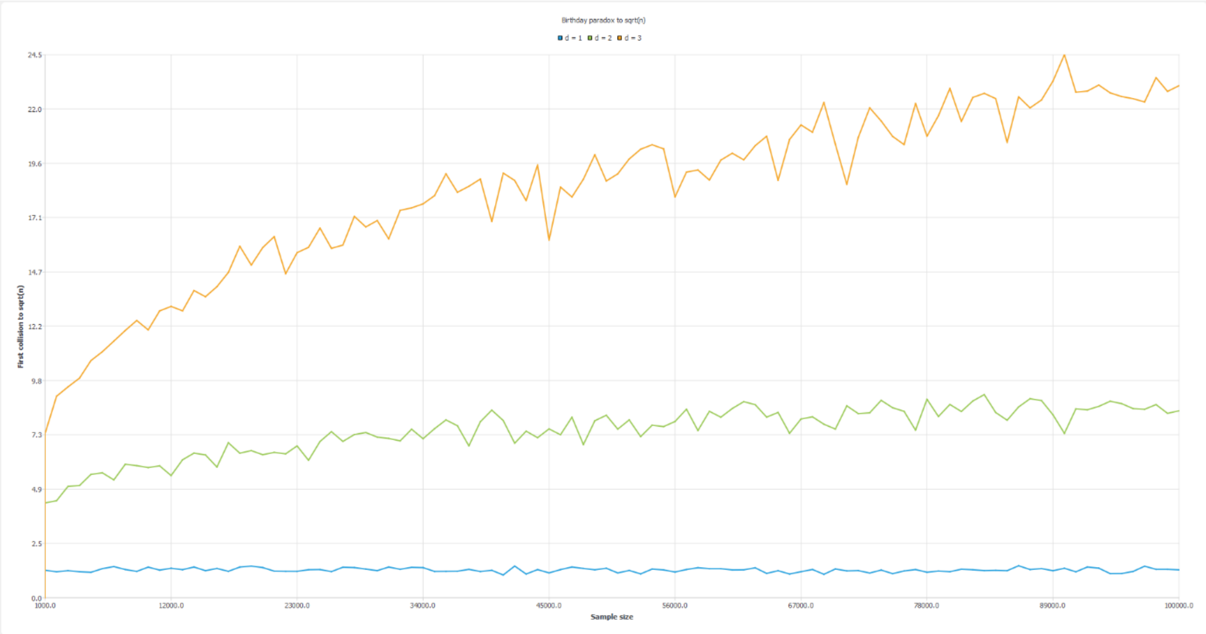
Dla każdej wrzucanej kuli wybieramy niezależnie i jednostajnie losowo d urn i umieszczamy kulę w najmniej zapełnionej z wybranych urn.

Poniżej porównanie wyników dla d = 1, 2, 3.

1. Moment pierwszej kolizji, pierwsza kula, która trafiła do niepustej urny (birthday paradox).

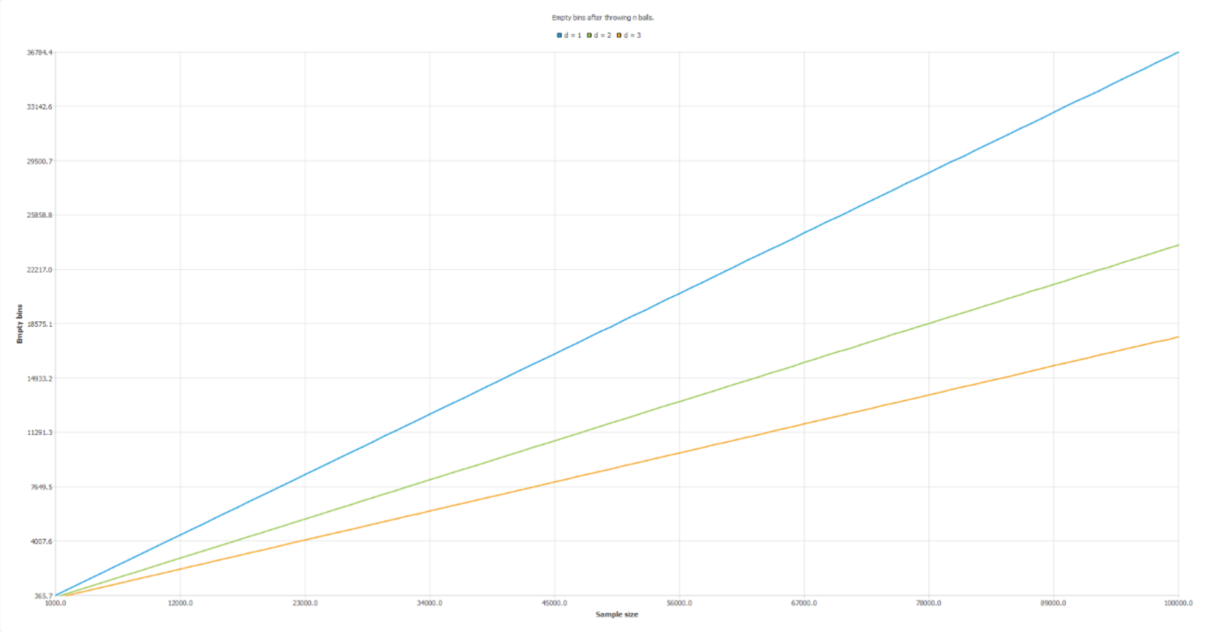






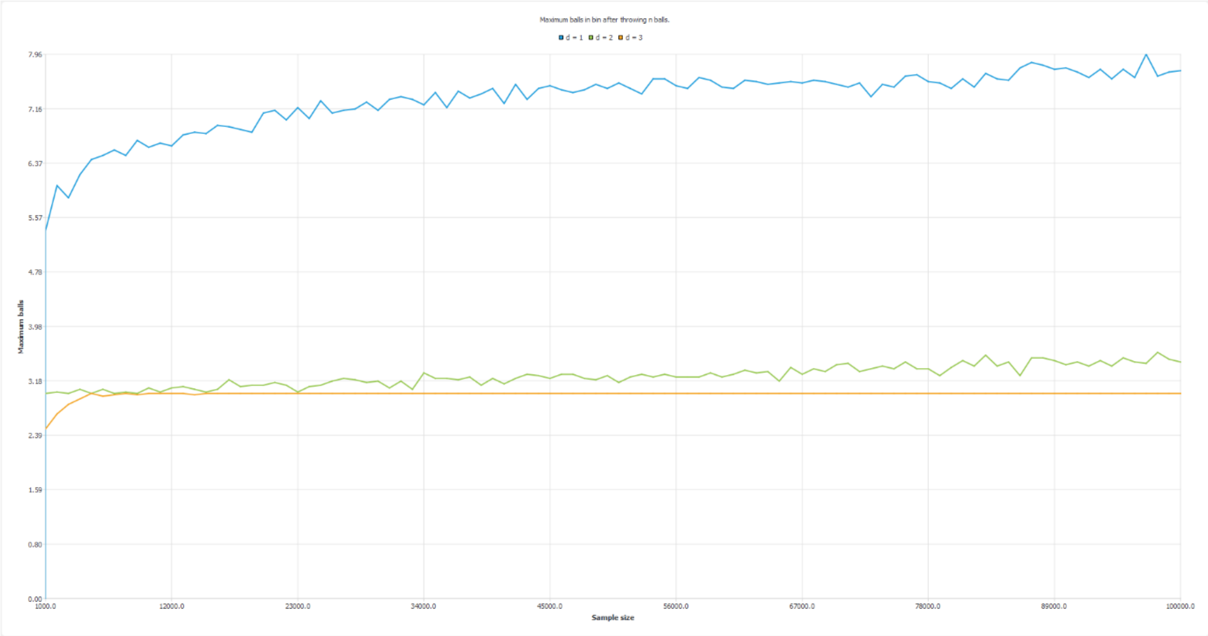
Wniosek: Wraz ze wzrostem d moment pierwszej kolizji staje się znacznie opóźniony – zgodnie z intuicją, unikamy kolizji losując więcej urn niż jedną.

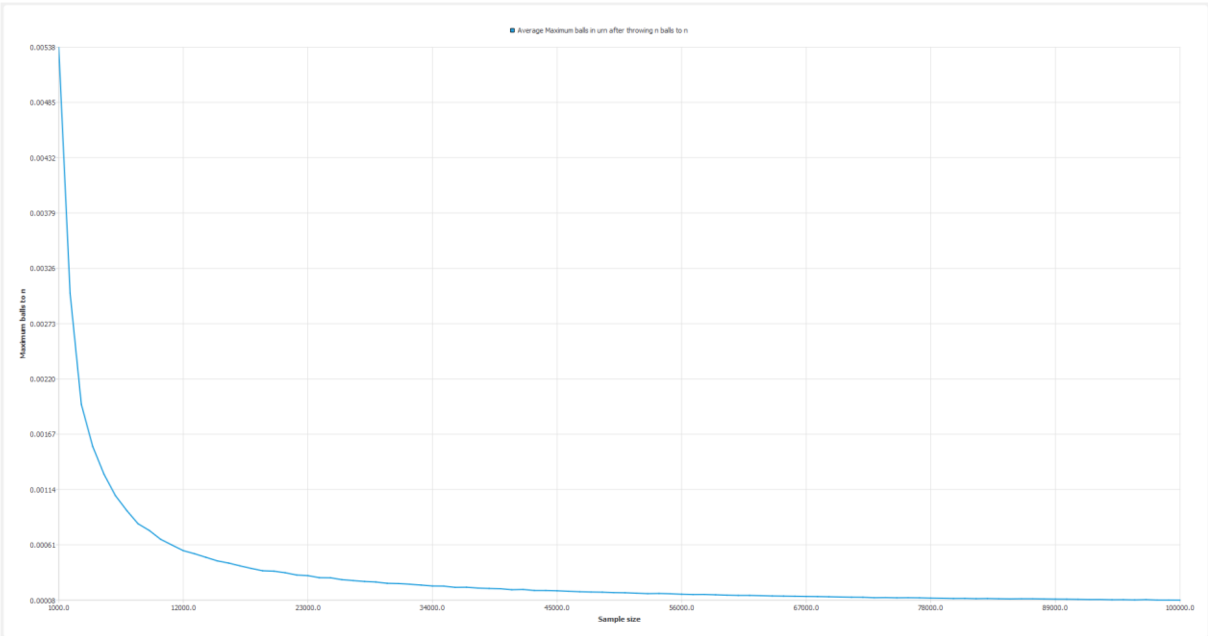
2. Liczba pustych urn po wrzuceniu n kul.

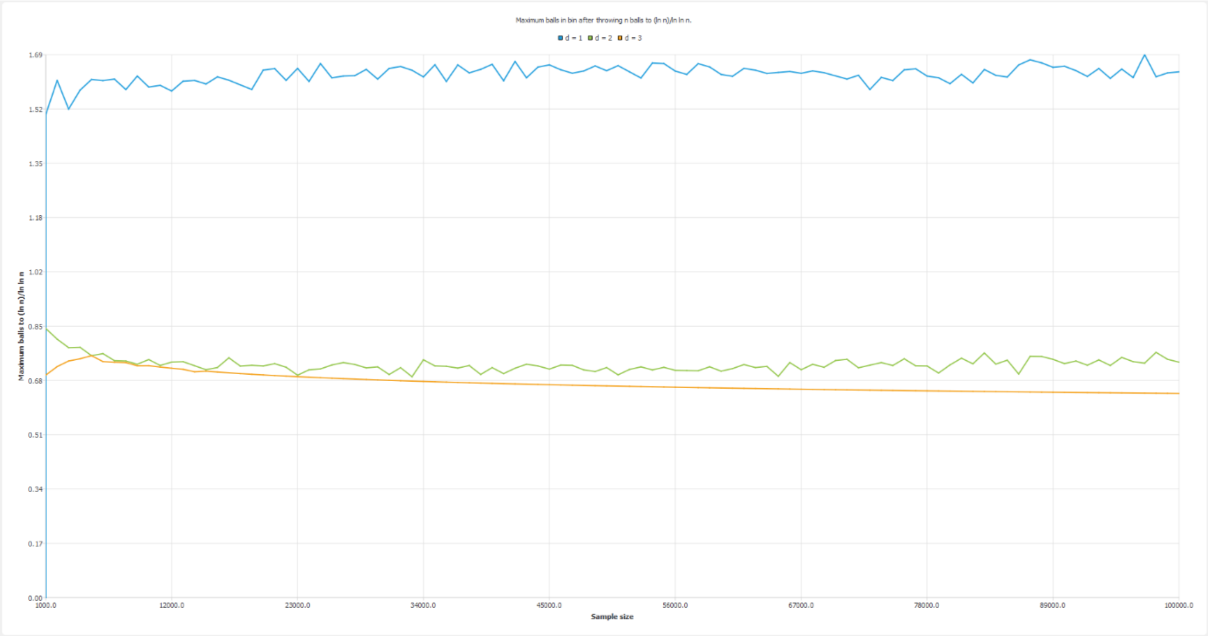


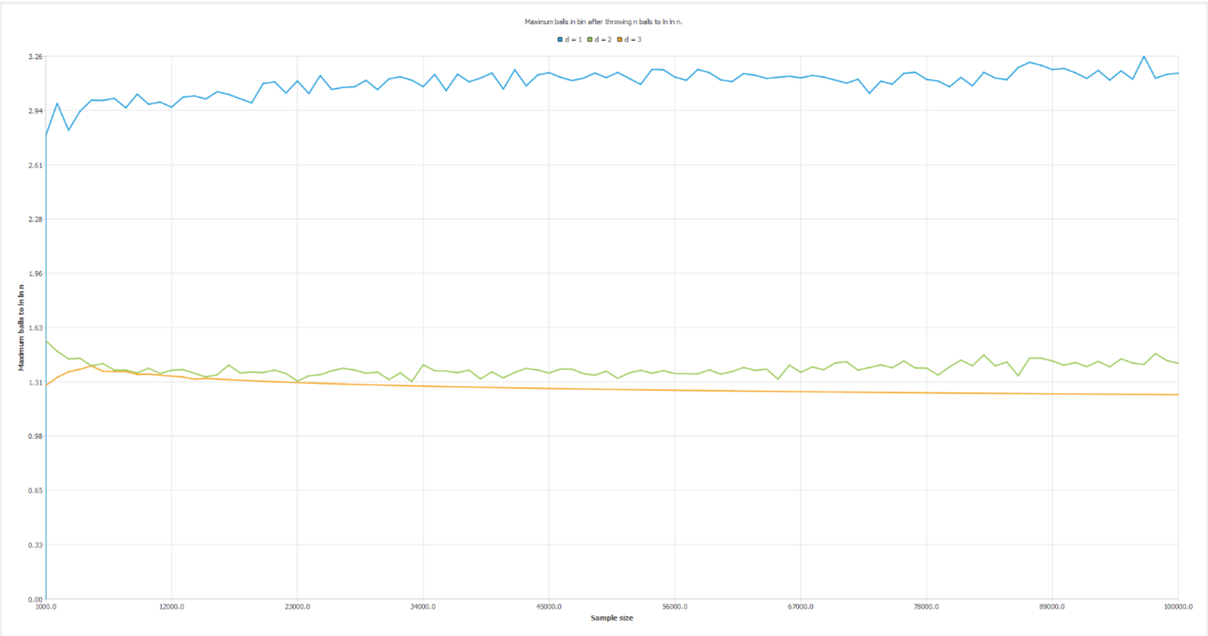
Wniosek: Dla większych d jest mniej pustych urn (logiczne – jeśli jedna z wybranych urn jest pusta, to do niej wrzucamy.

3. Maksymalna liczba kul w urnie po wrzuceniu n kul.



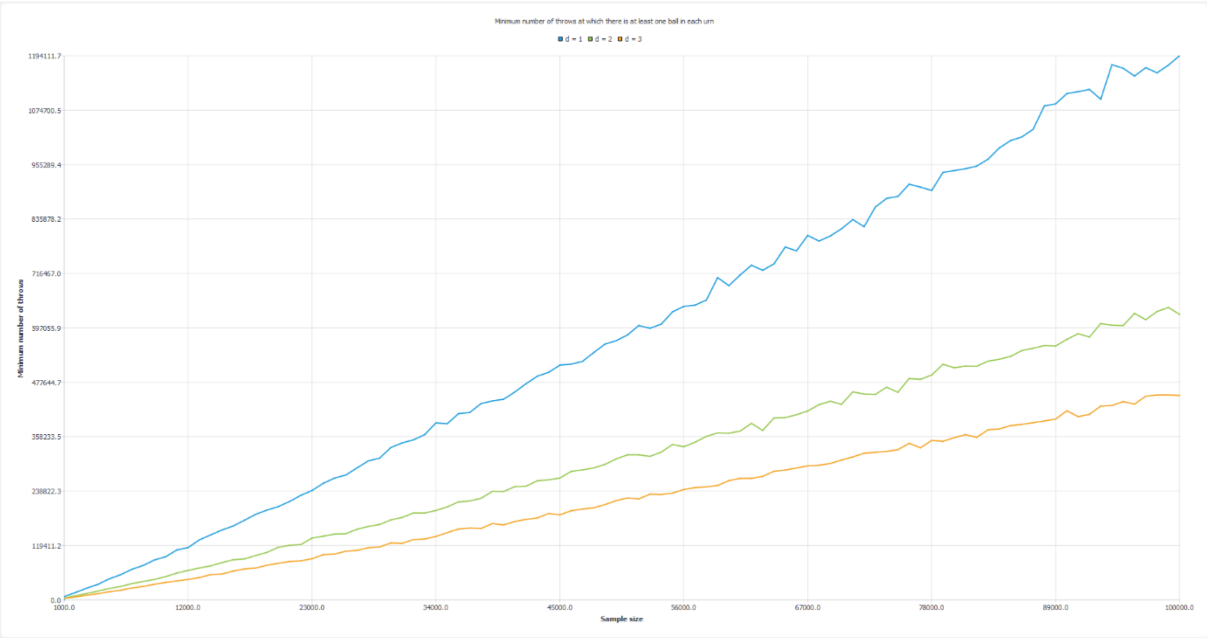


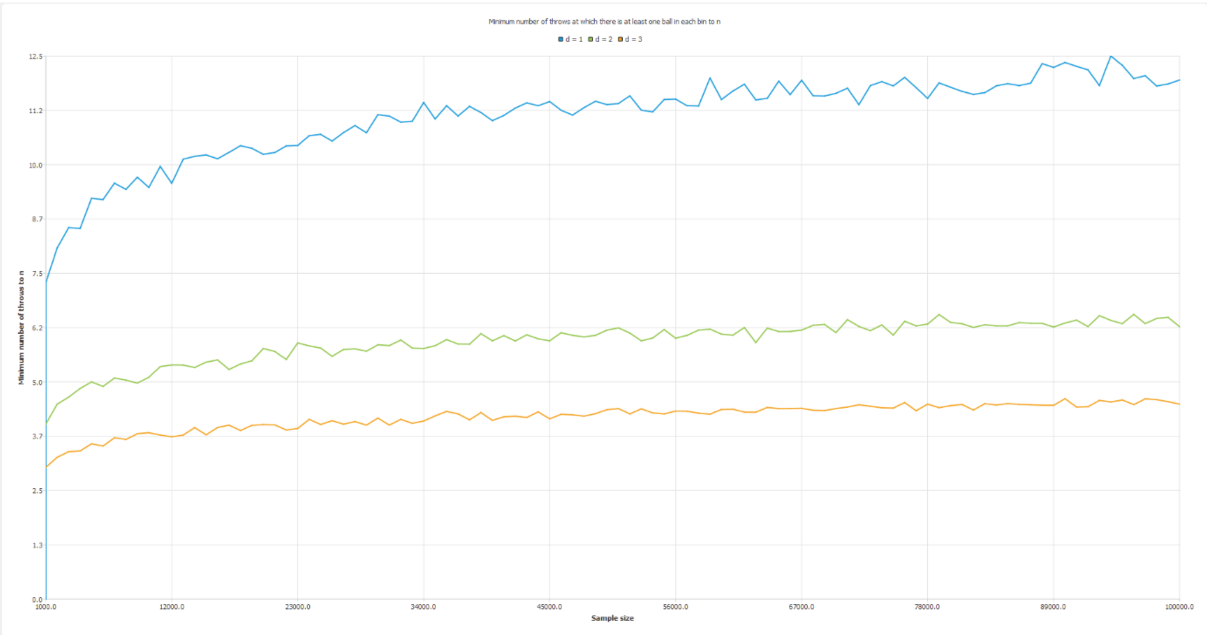


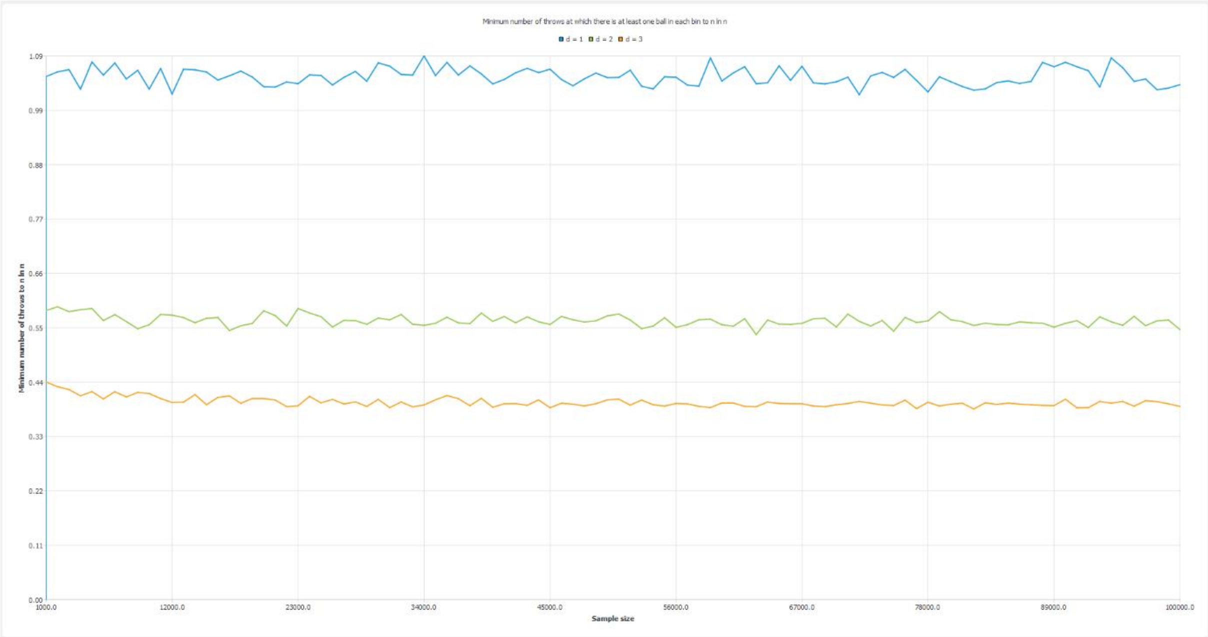


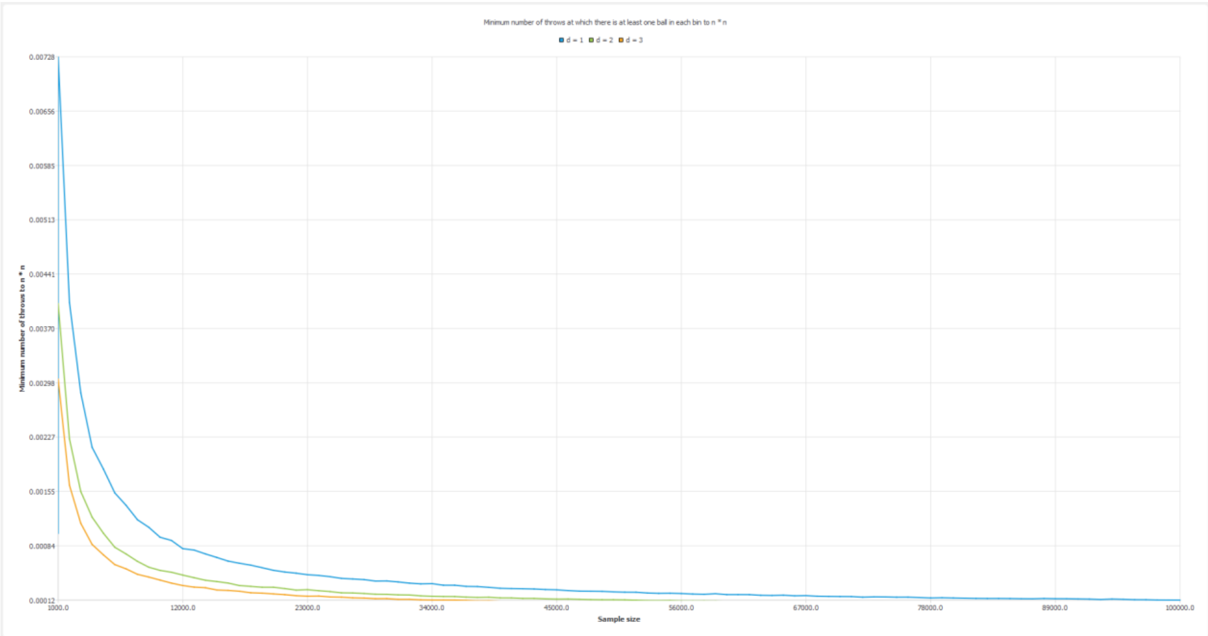
Wniosek: Średnia maksymalna liczba kul w urnie po wrzuceniu n kul dla d = 1 oscyluje wokół 1.6*((ln(n))/(ln(ln(n))), dla d = 2 oscyluje wokół 0.7*((ln(n))/(ln(ln(n))) (bardzo powoli rośnie, minimum 3 kule), dla d = 3 dla dużych n (n > 12 000) wynosi 3 kule.

4. Minimalna liczba rzutów, po której w każdej z urn jest co najmniej jedna kula (problem kolekcjonera kuponów).



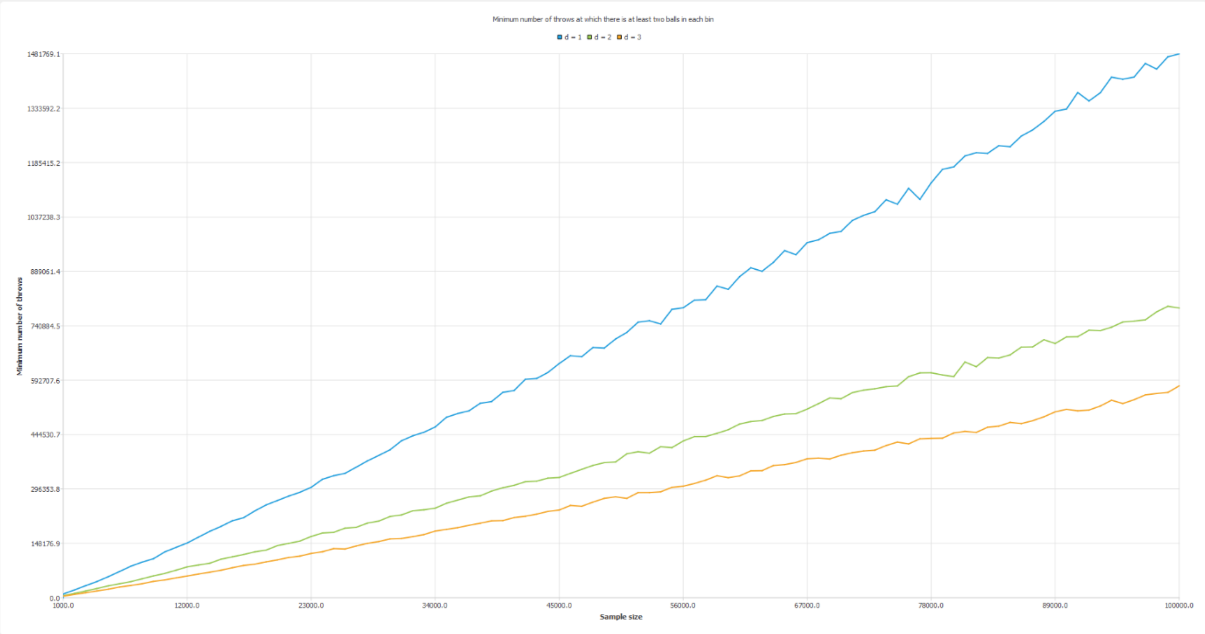


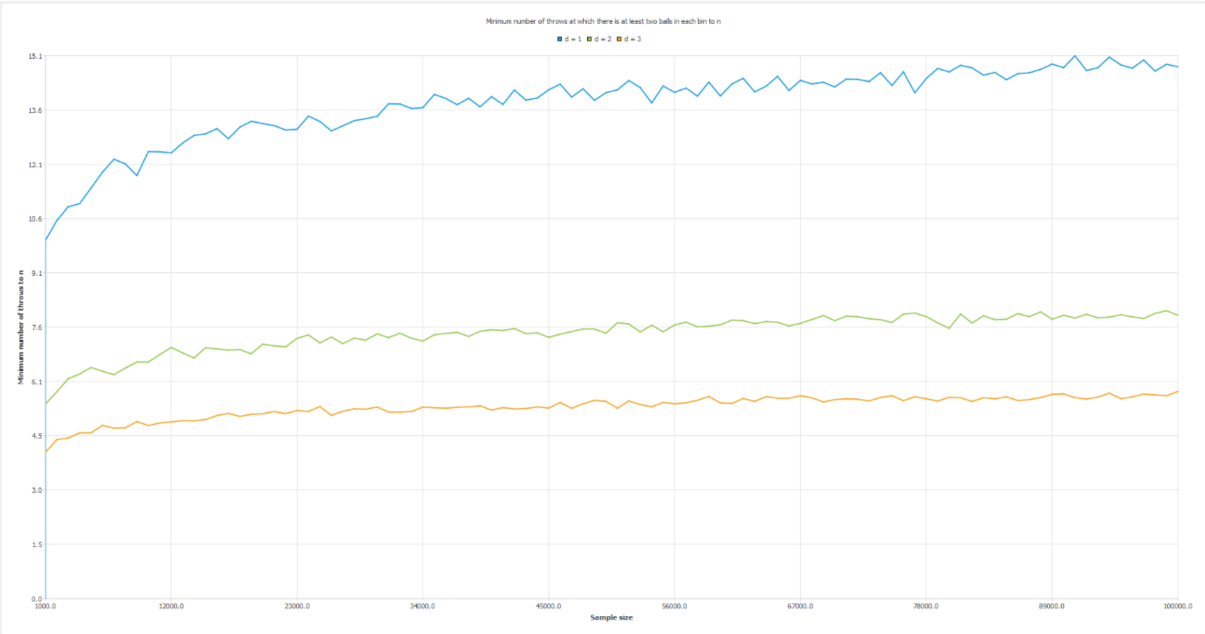


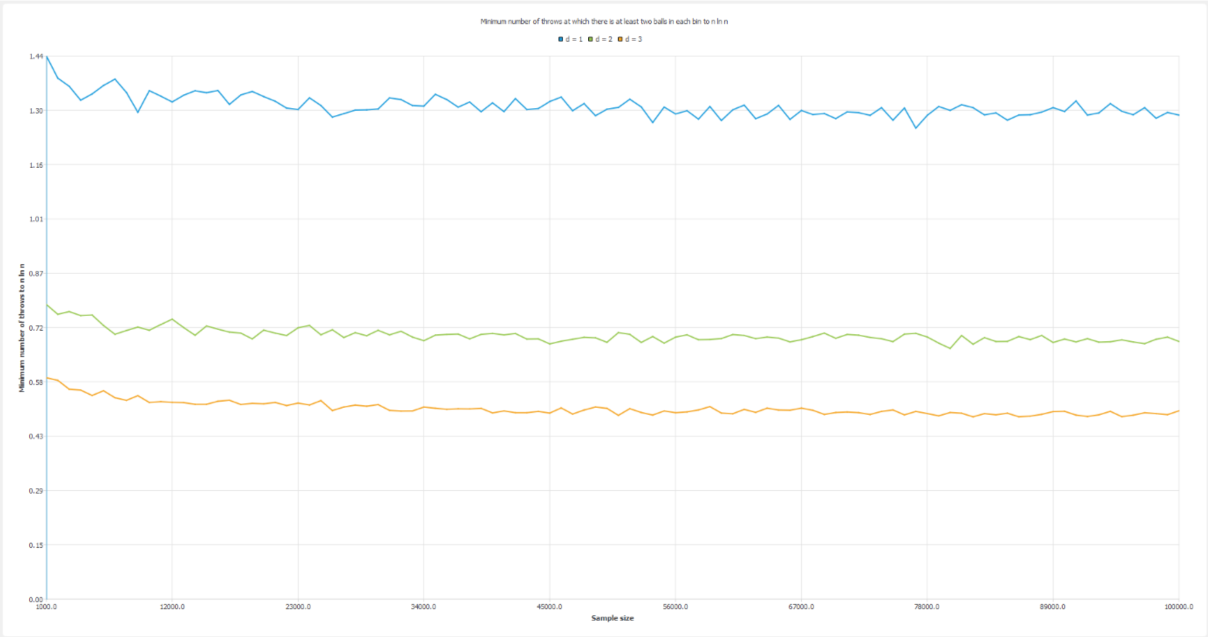


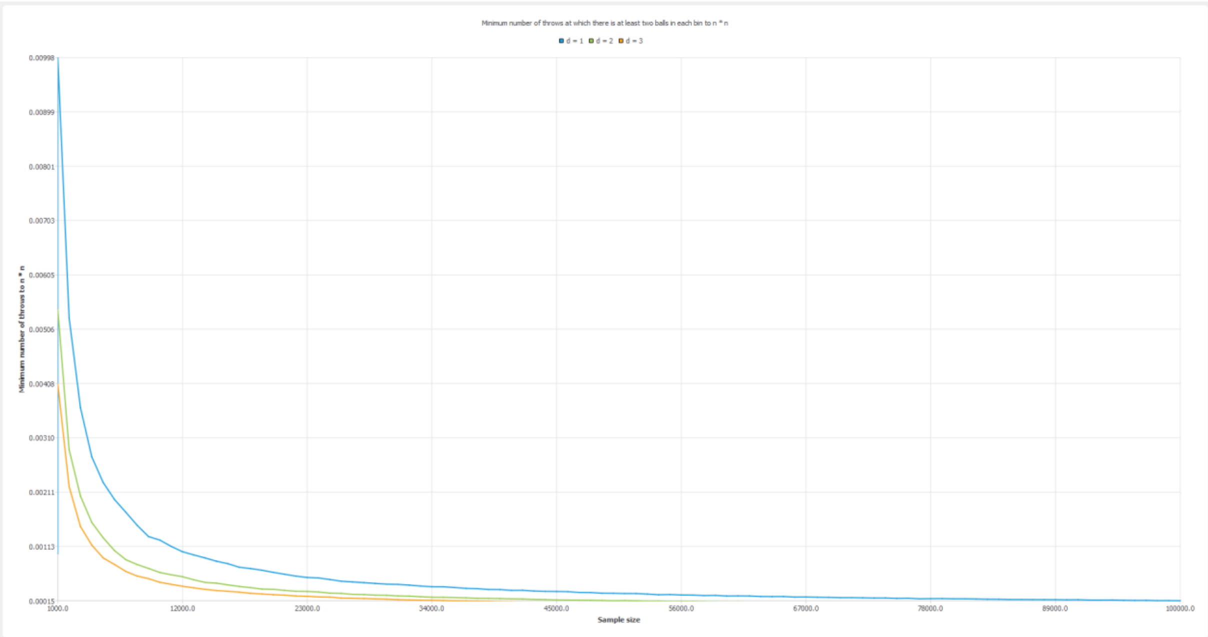
Wniosek: Średnia minimalna liczba rzutów, po której w każdej z urn jest co najmniej jedna kula dla d=1 oscyluje wokół 1.05*n*ln(n), dla d=2 oscyluje wokół 0.6*n*ln(n),a dla d=3 oscyluje wokół 0.4**n*ln(n). Analogicznie do przypadku pustych urn, zwiększenie d gwarantuje bardziej równomierne rozkładanie kul do urn(zwiększa prawdopodobieństwo skolekcjonowania wszystkich kuponów).

5. Minimalna liczba rzutów, po której w każdej z urn są co najmniej dwie kule.





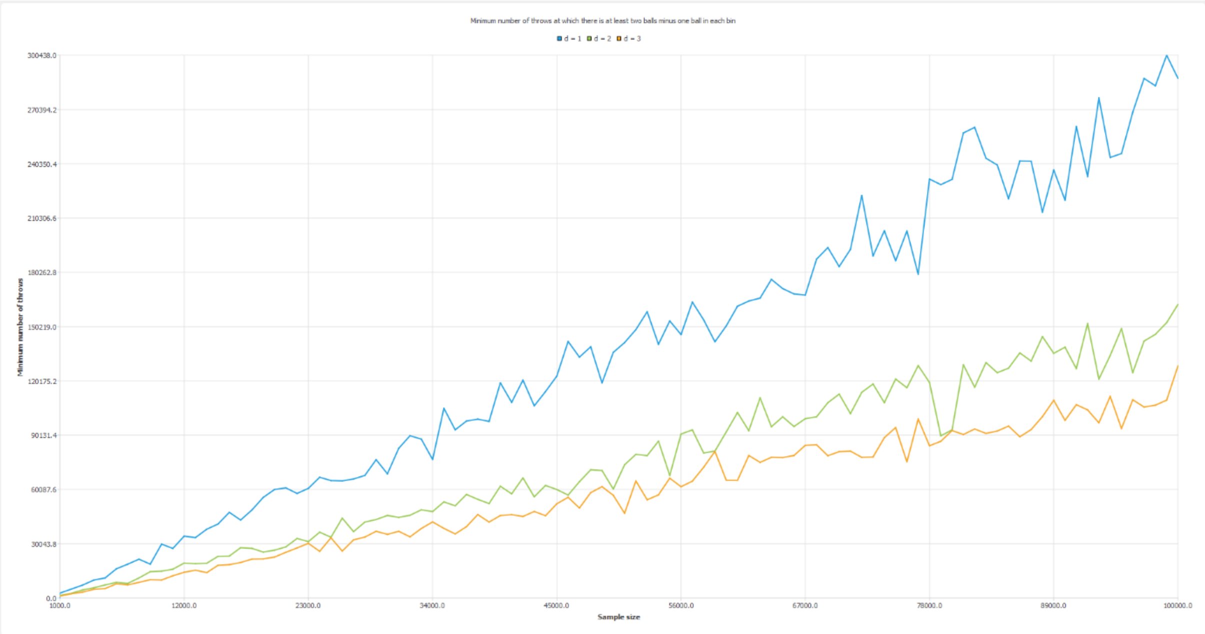


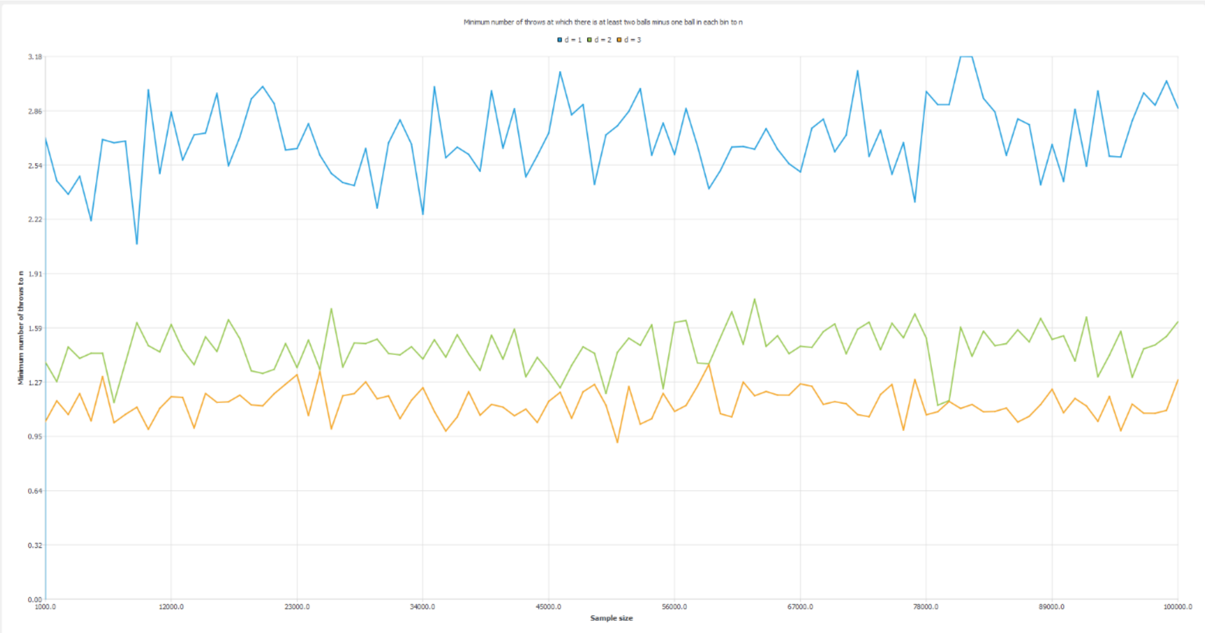


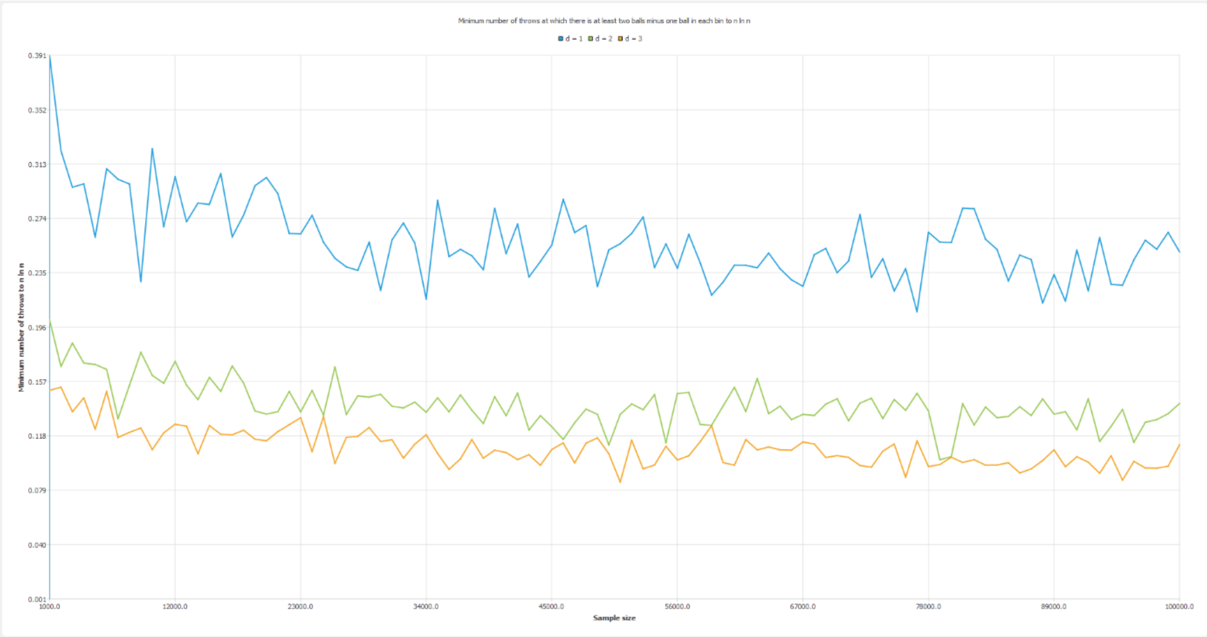
Wniosek: Średnia różnica pomiędzy minimalną liczbą rzutów, po której w każdej z urn są conajmniej dwie kule a minimalną liczbą rzutów, po której w każdej z urn jest conajmniej jedna kula dla d=1 oscyluje wokół 1.3*n*ln(n), dla d=2 oscyluje wokół 0.7*n*ln(n), dla d=3 oscyluje wokół 0.5*n*ln(n).Tak jak dla poprzedniego przypadku zwiększenie d sprawia, że po mniejszej liczbie rzutów w każdej z urn są co najmniej dwie kule.

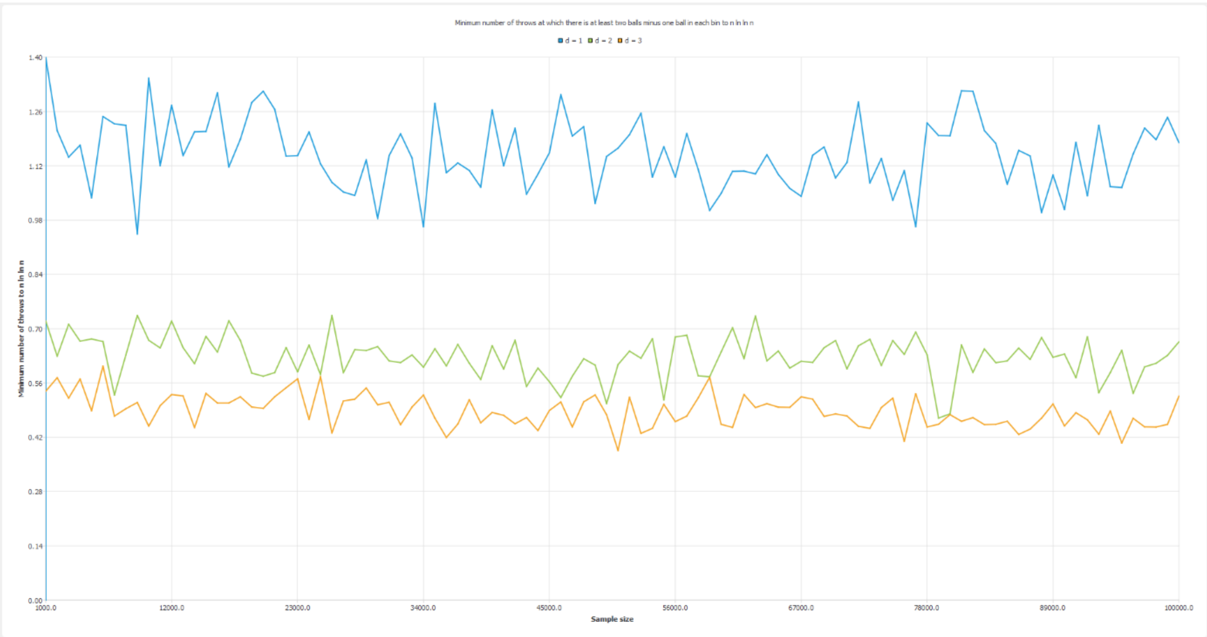
Zwiększenie d powoduje bardziej równomierny rozkład kul do urn.

6. Różnica pomiędzy minimalną liczbą rzutów, po której w każdej z urn są co najmniej dwie kule a minimalną liczbą rzutów, po której w każdej z urn jest co najmniej jedna kula.









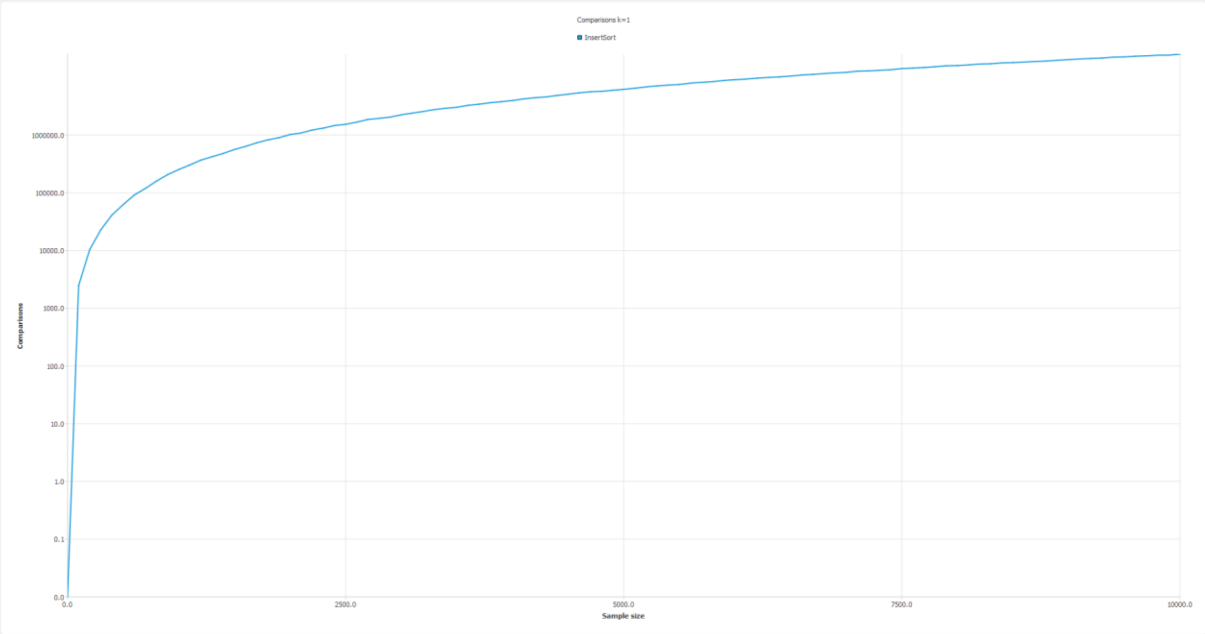
Wniosek: Czym d jest większe tym mniejsza jest różnica pomiędzy minimalną liczbą rzutów, po której w każdej z urn są co najmniej dwie kule a minimalną liczbą rzutów, po której w każdej z urn jest co najmniej jedna kula. Kule są bardziej równomiernie rozłożone.

Zad. 3

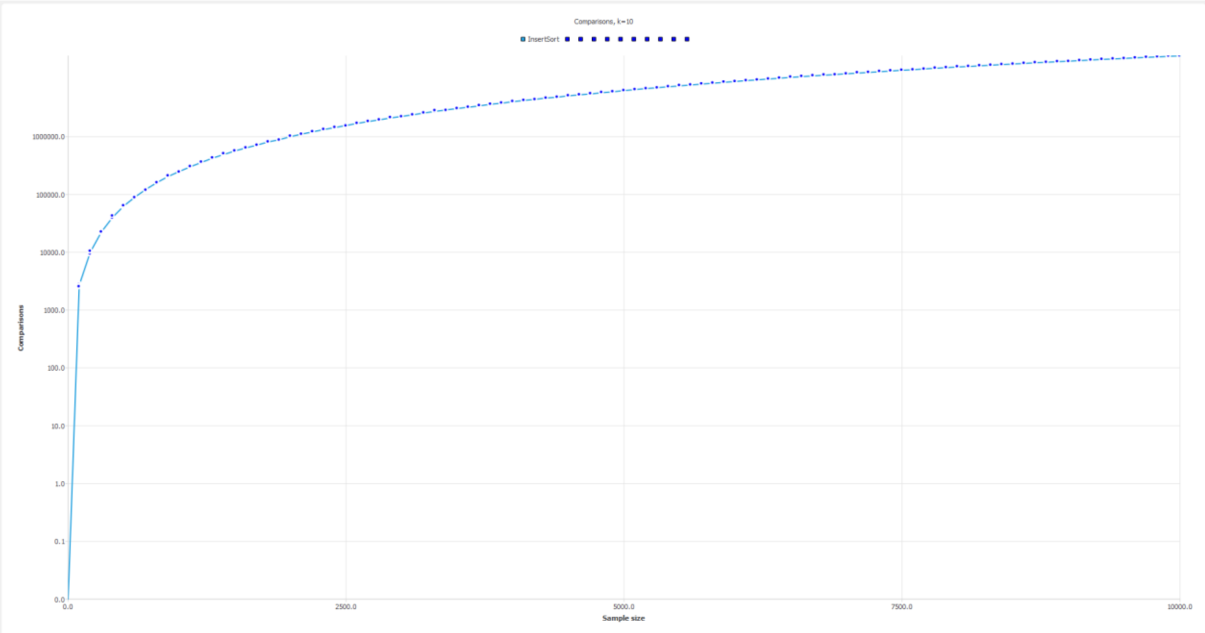
Testowanie działania algorytmu sortowania przez wstawianie InsertionSort dla k niezależnych powtórzeń.

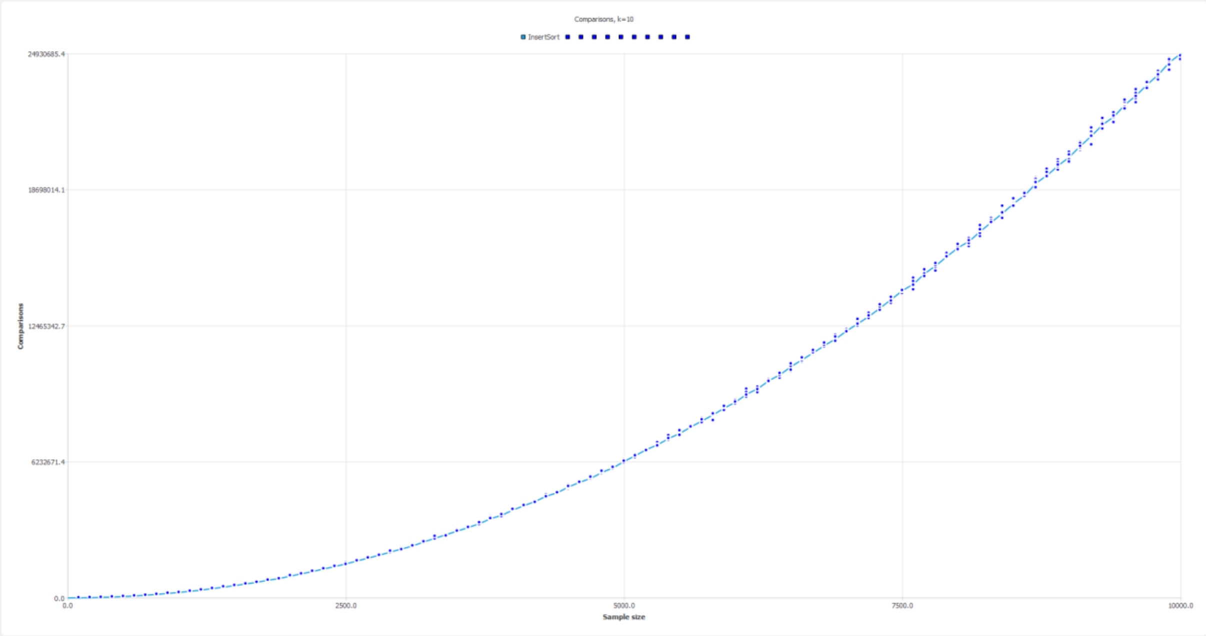
1. Liczba wykonanych porównań w poszczególnych powtórzeniach oraz średnia liczba porównań cmp(n) w zależności od n.

k = 1



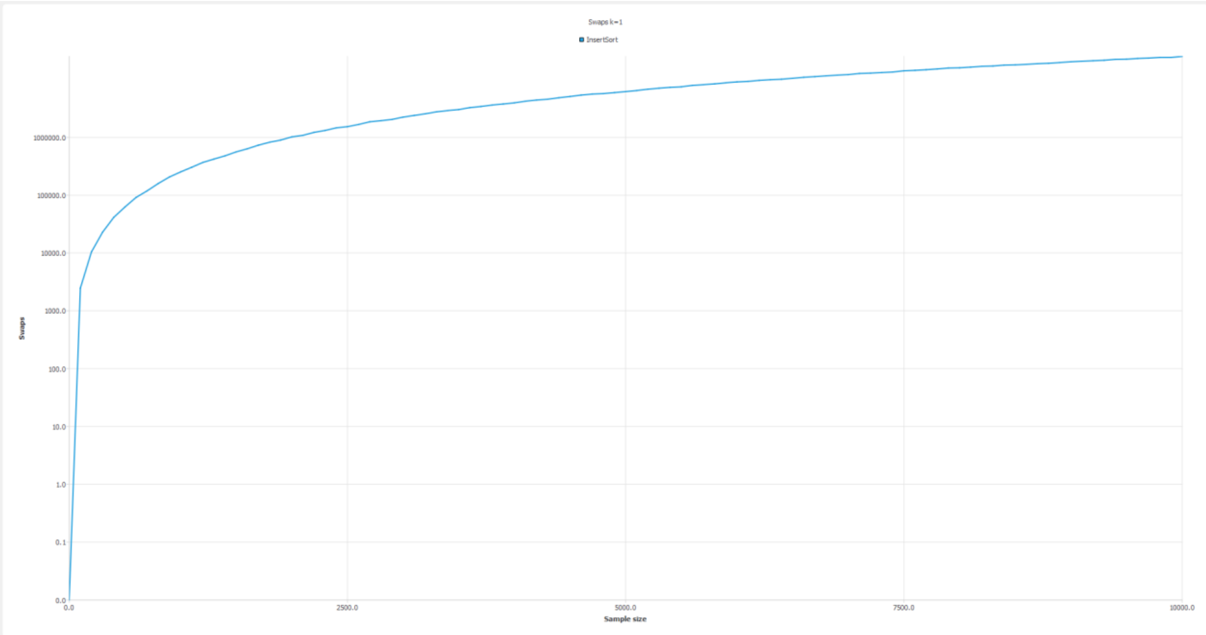
k = 10



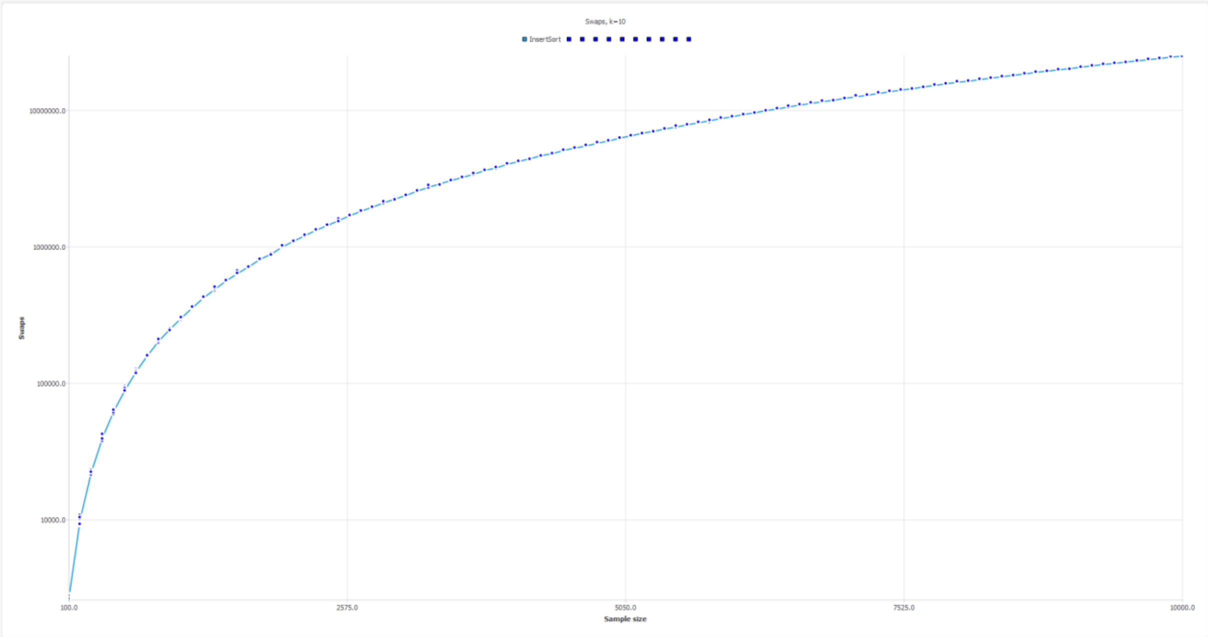


2. Liczba przestawień kluczy w poszczególnych powtórzeniach oraz średnia liczba przestawień s(n) w zależności od n.

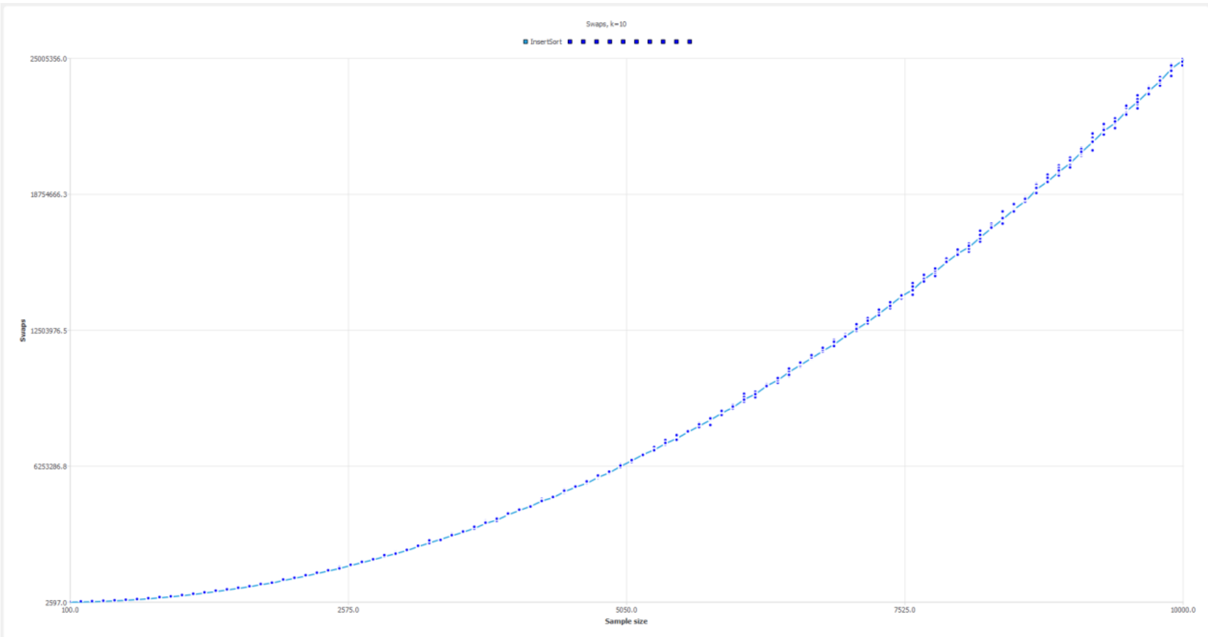
k = 1



k = 10

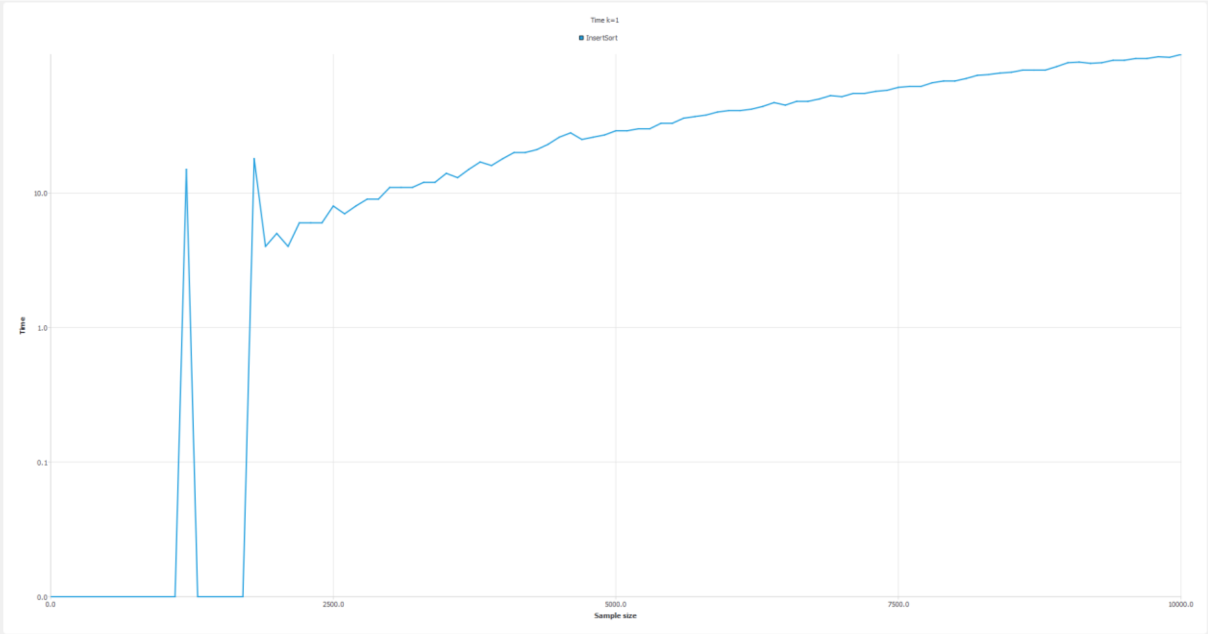

Swaps, k=10
InsertSort
10000000.0
1000000.0
100000.0
10000.0
Swaps
100.0
2575.0
5050.0
7525.0
10000.0
Sample size

k = 10

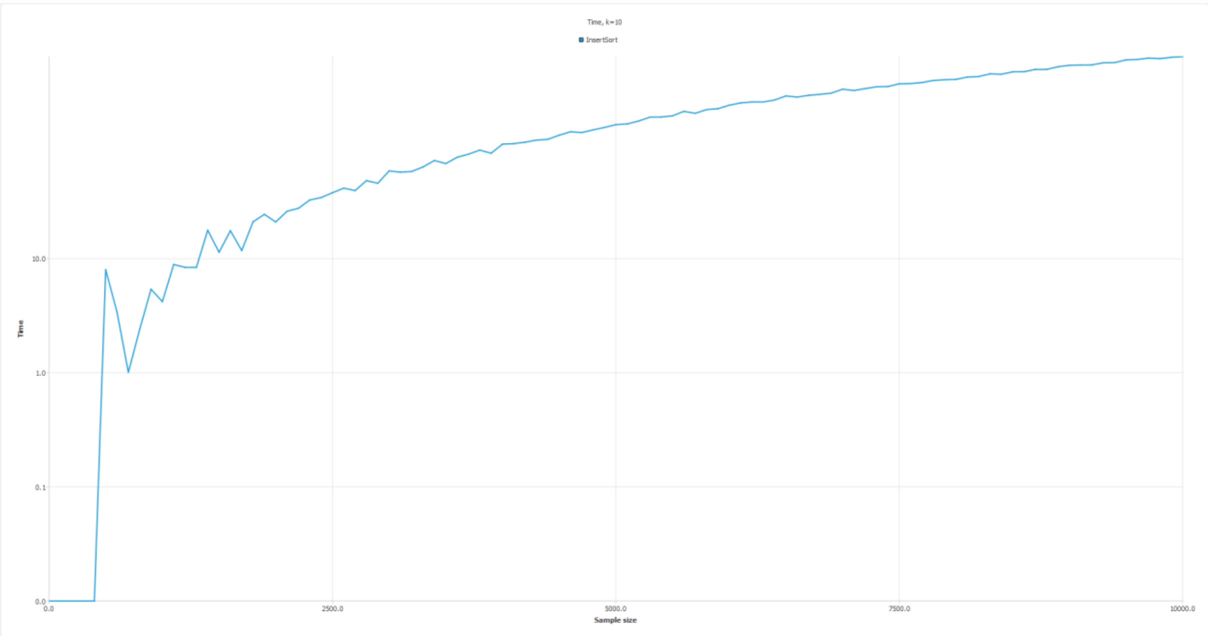

Swaps, k=10
InsertSort
25005356.0
18754666.3
12503976.5
6253286.8
2597.0
Swaps
100.0
2575.0
5050.0
7525.0
10000.0
Sample size

3. Czas działania dla poszczególnych powtórzeń oraz średni czas działania algorytmu w zależności od n. Widać anomalie spowodowane bardzo małym czasem działania programu dla małych próbek (nie udało się zmierzyć czasu działania programu a ten już się skończył).

k = 1



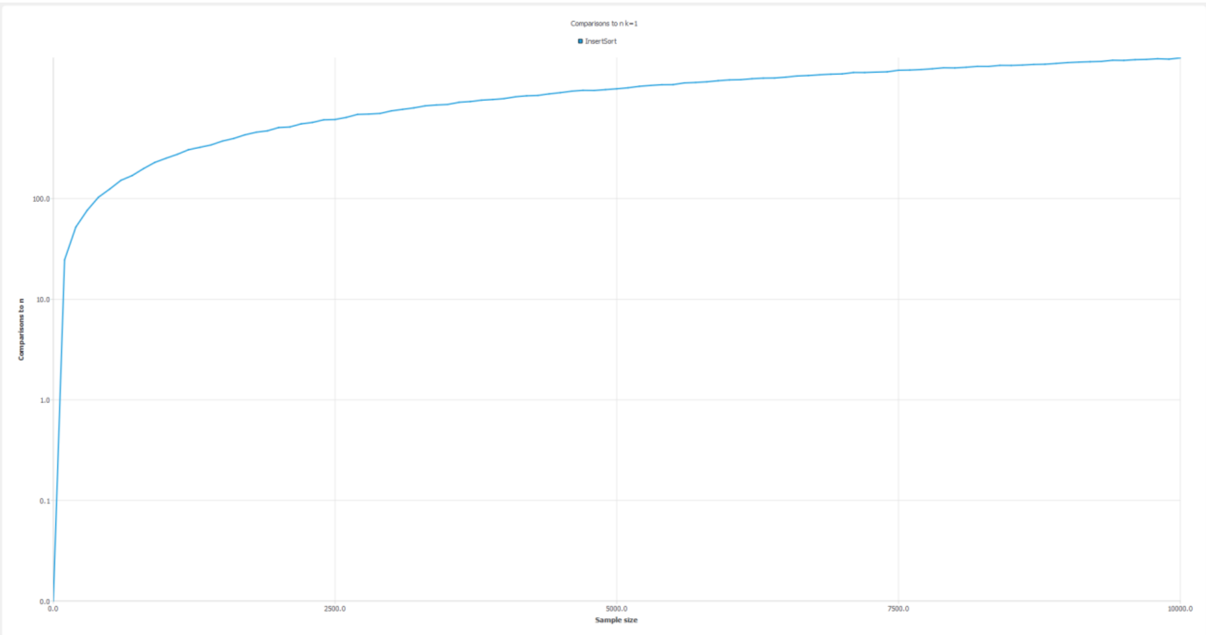
k = 10



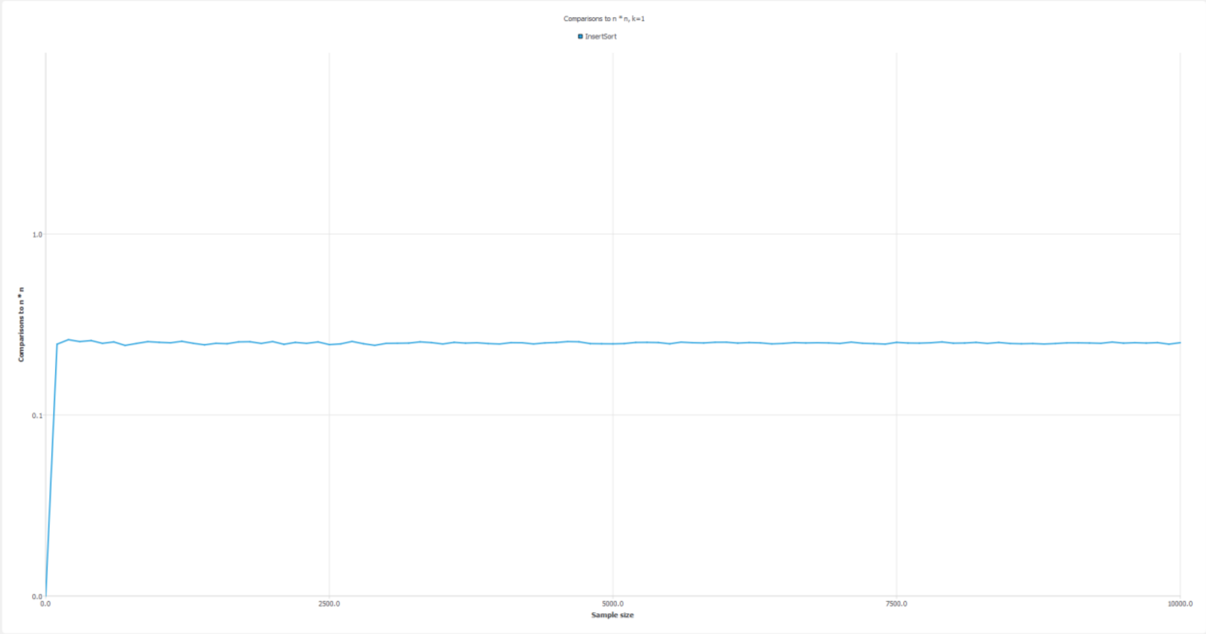
k = 10

4. Iloraz cmp(n)/n oraz cmp(n)/n2

k = 1

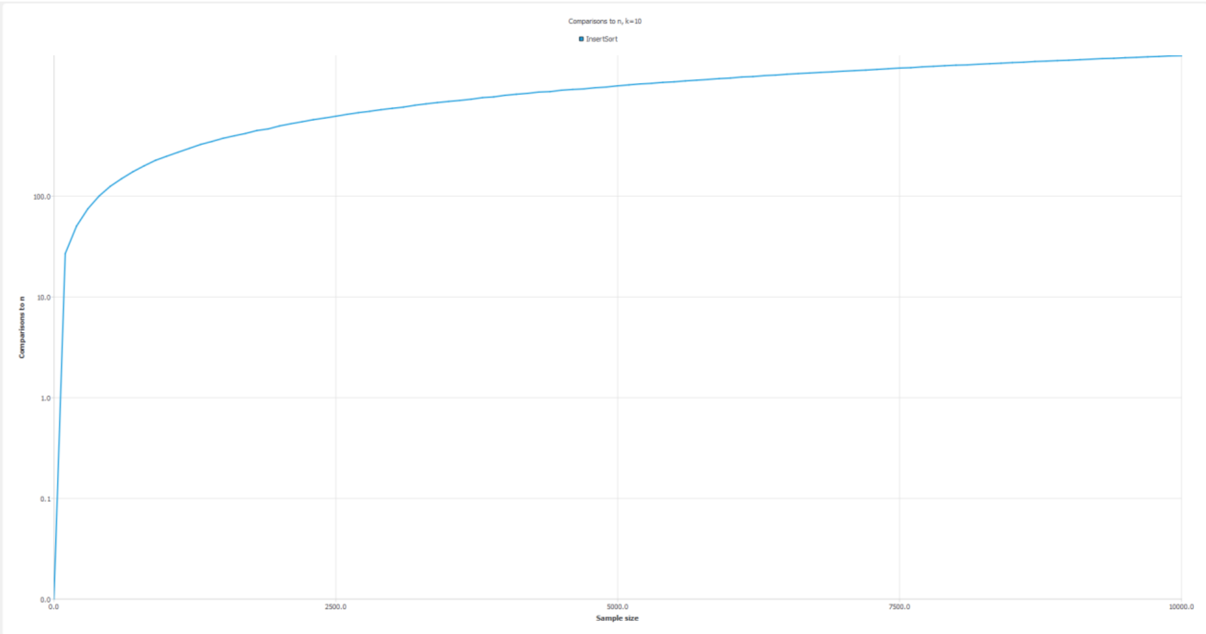


k = 1



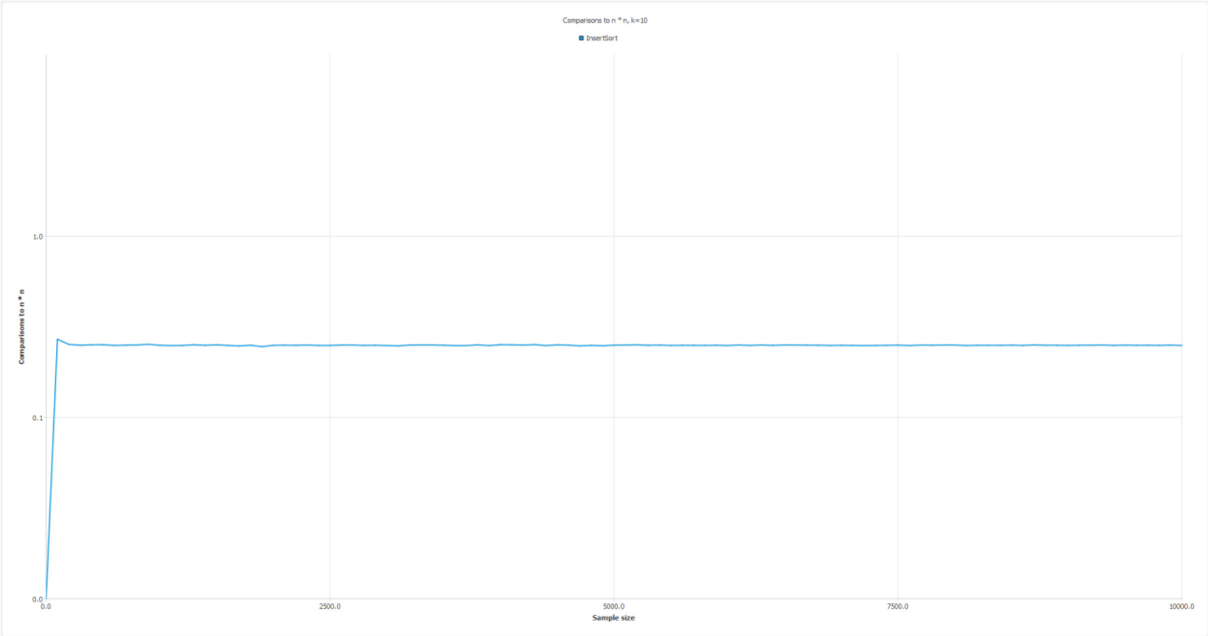
Comparisons to n * n, k=1
InsertSort
Comparisons to n * n
1.0
0.1
0.0
0.0
2500.0
5000.0
7500.0
10000.0
Sample size

k = 10



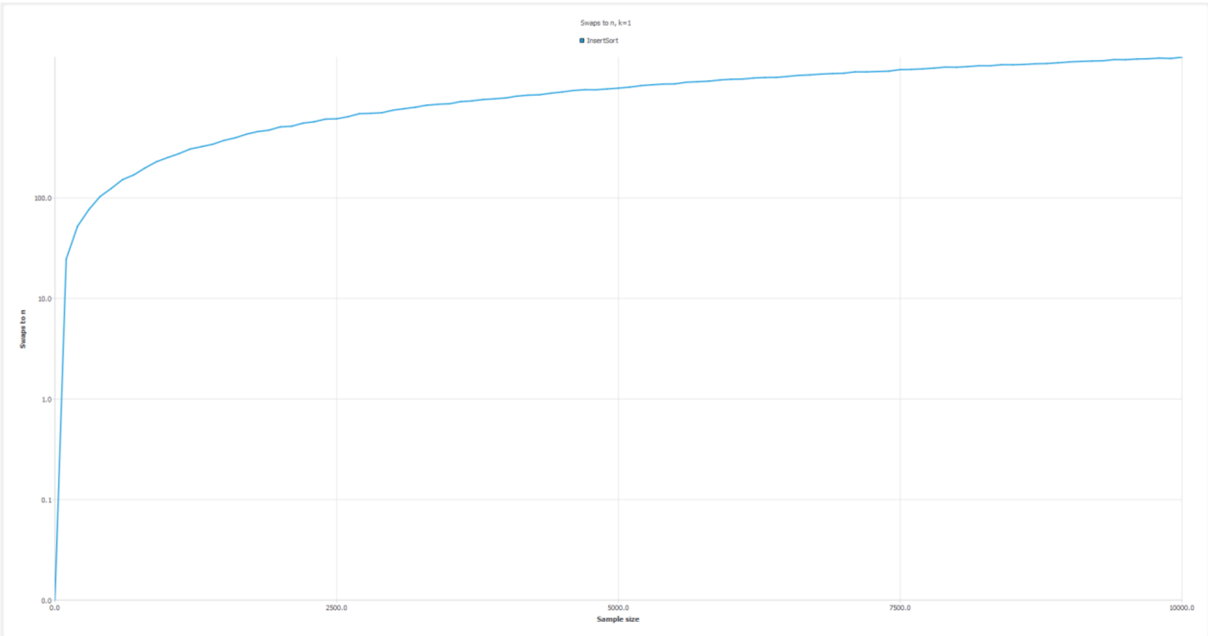
Comparisons to n, k=10
InsertSort
Comparisons to n
100.0
10.0
1.0
0.1
0.0
0.0
2500.0
5000.0
7500.0
10000.0
Sample size

k = 10

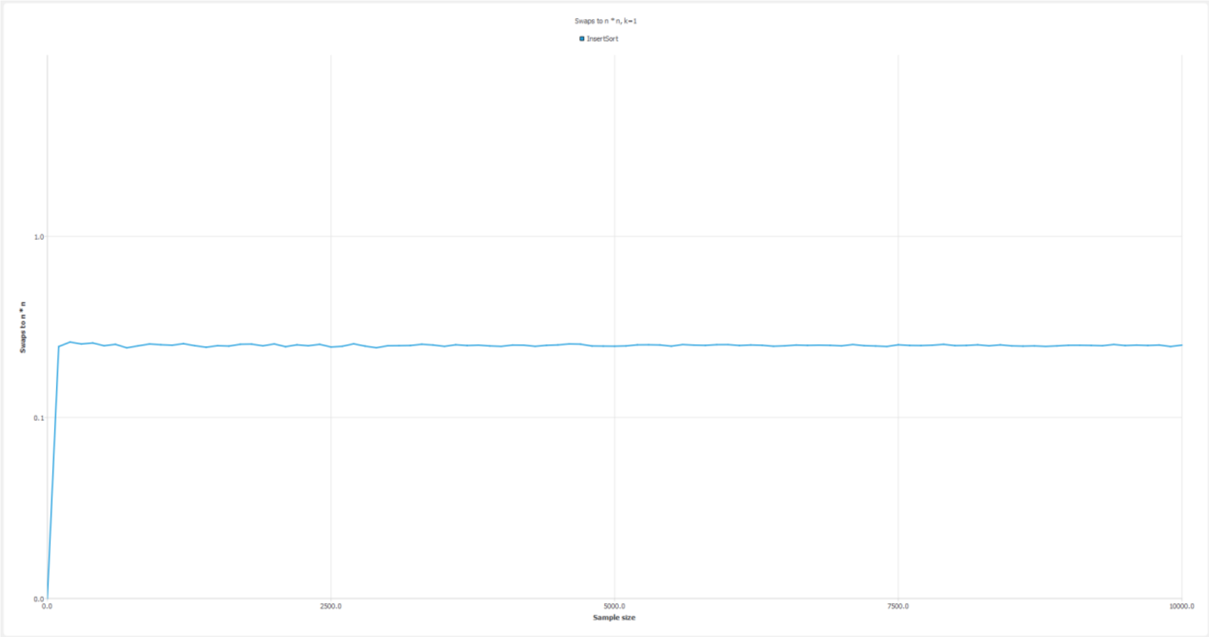


5. Iloraz s(n)/n oraz s(n)/n2

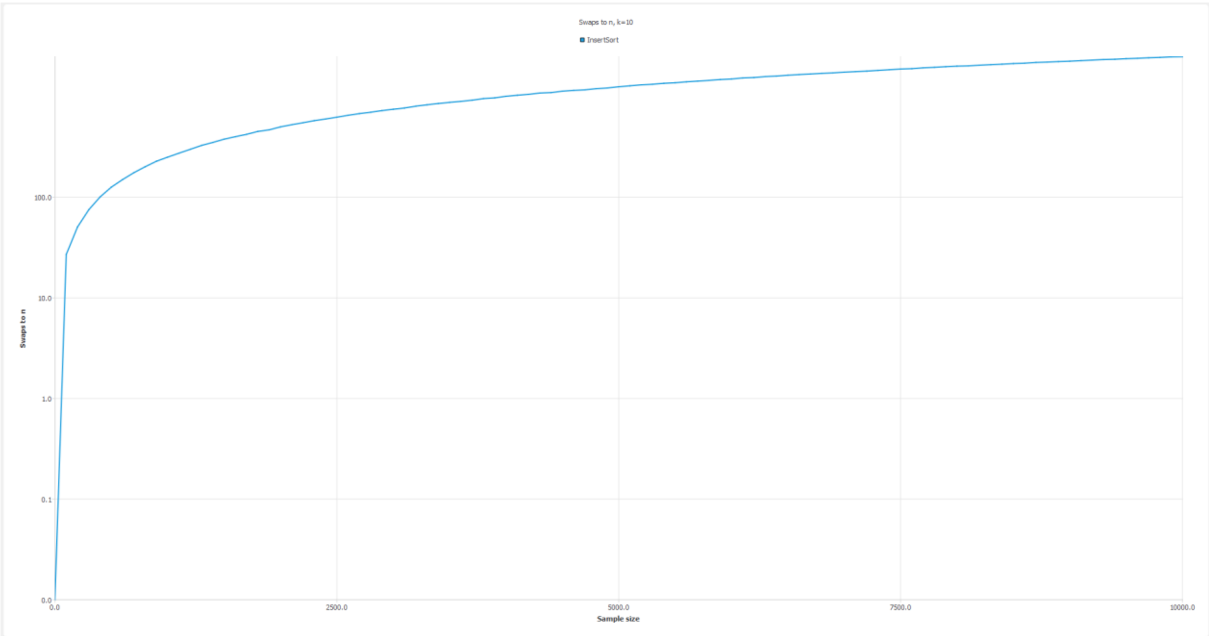
k = 1



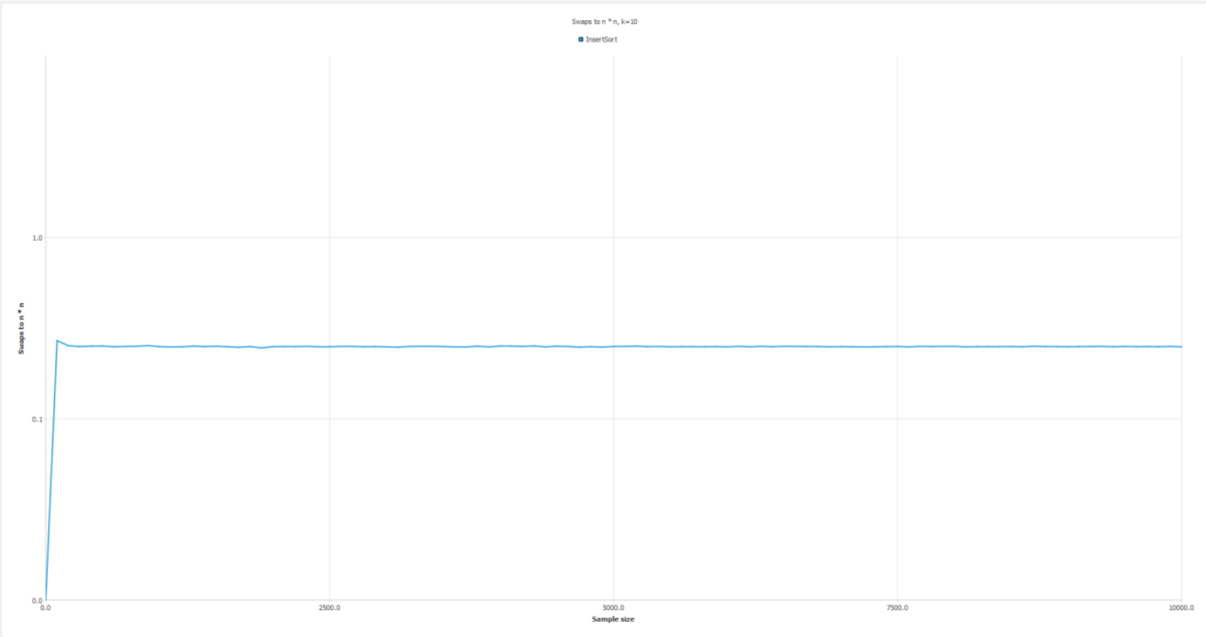
k = 1


Swaps to n * n, k=1
InsertSort
Swaps to n * n
1.0
0.1
0.0
0.0
2500.0
5000.0
7500.0
10000.0
Sample size

k = 10


Swaps to n, k=10
InsertSort
Swaps to n
100.0
10.0
1.0
0.1
0.0
0.0
2500.0
5000.0
7500.0
10000.0
Sample size

k = 10



Swaps to n * n, k=10
InsertSort
Swaps to n * n
1.0
0.1
0.0
0.0
2500.0
5000.0
7500.0
10000.0
Sample size