Zadanie 1

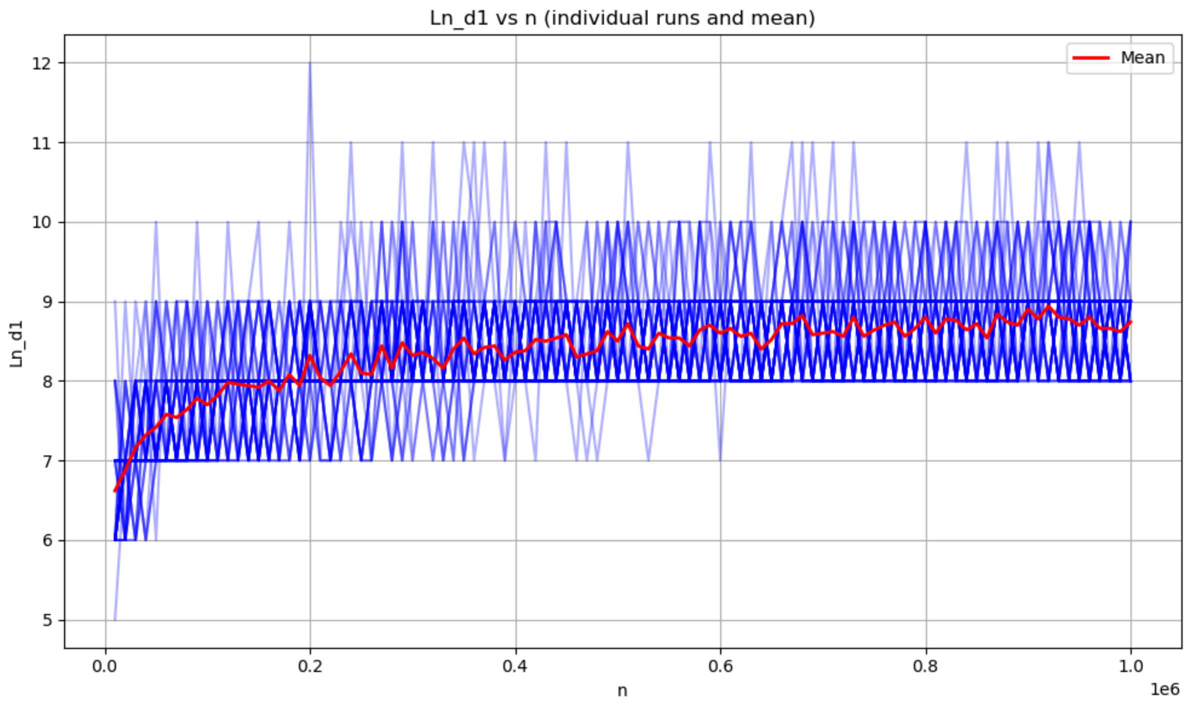
Cel eksperymentu

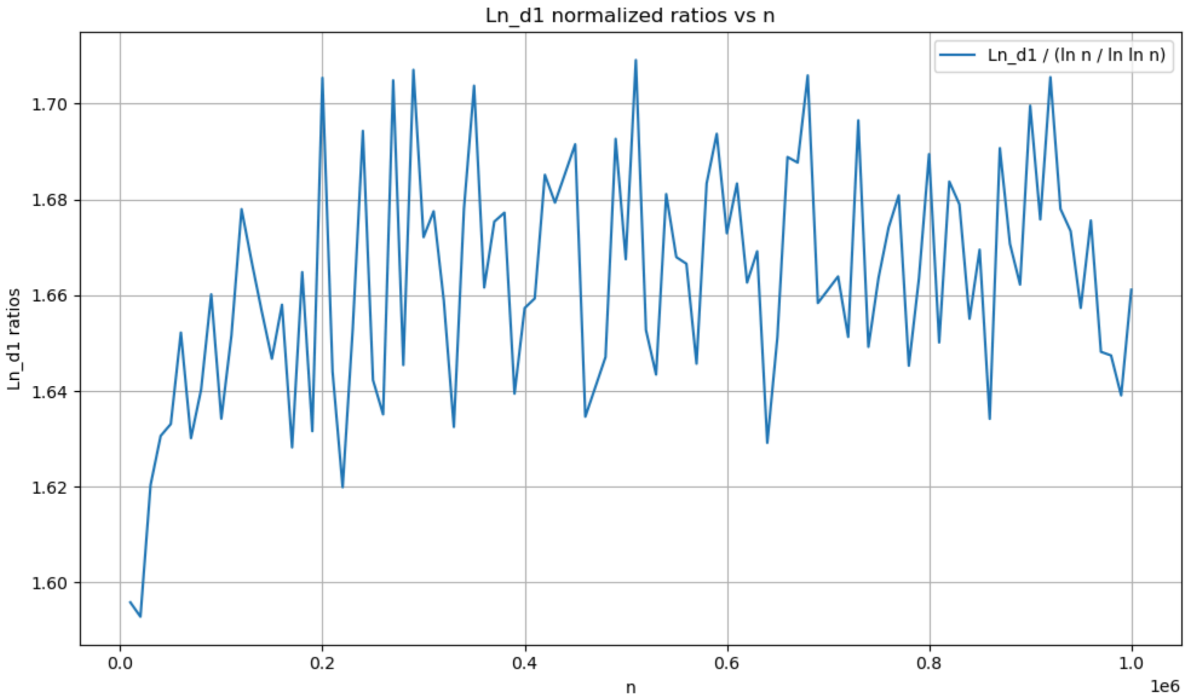
Celem eksperymentu było zbadanie maksymalnego zapełnienia pojedynczej urny po wrzuceniu n kul do n urn w dwóch scenariuszach:

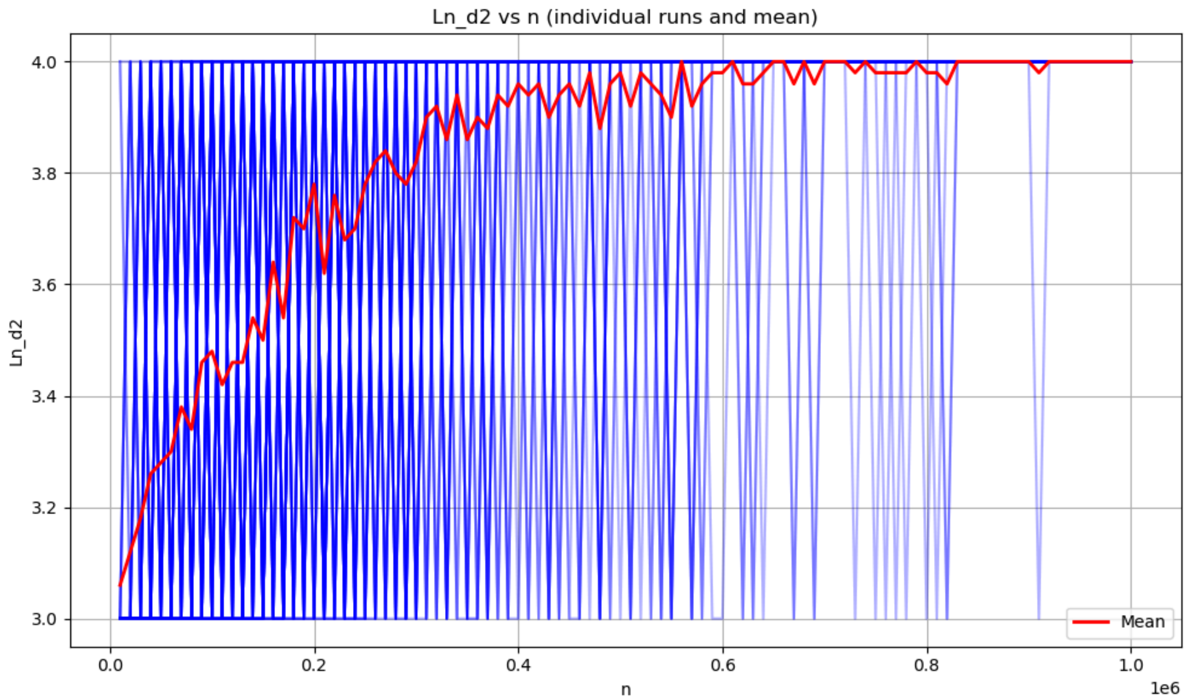
1. $\boldsymbol{d = 1}$: każda kula jest wrzucana do jednej losowo wybranej urny.
2. $\boldsymbol{d = 2}$: każda kula jest wrzucana do dwóch losowo wybranych urn, a następnie trafia do urny z mniejszą liczbą kul (w przypadku remisu wybór jest dowolny).

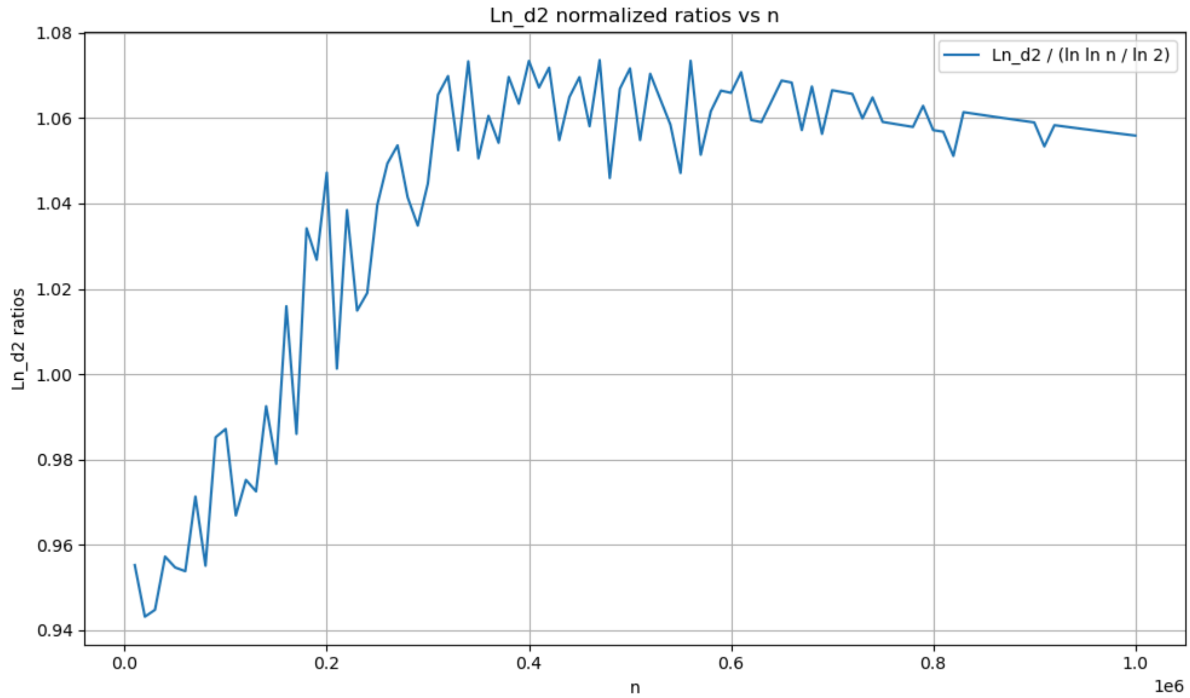
Eksperyment miał na celu sprawdzenie, jak zastosowanie strategii **Power of Two Choices** wpływa na maksymalną liczbę kul w jednej urnie, oraz weryfikację teoretycznych przewidywań asymptotycznych.

Wykresy




Ln_d1 normalized ratios vs n
Ln_d1 / (ln n / ln ln n)
Ln_d1 ratios
1.60
1.62
1.64
1.66
1.68
1.70
0.0
0.2
0.4
0.6
0.8
1.0
1e6
n


Ln_d2 vs n (individual runs and mean)
Mean
Ln_d2
3.0
3.2
3.4
3.6
3.8
4.0
0.0
0.2
0.4
0.6
0.8
1.0
1e6
n



Metoda

Dla każdego n w zakresie od 10 000 do 1 000 000 wykonano $k = 50$ niezależnych powtórzeń eksperymentu dla obu scenariuszy. W każdym powtórzeniu zliczano liczbę kul w każdej urnie po wrzuceniu wszystkich n kul i zapisywano maksymalne zapełnienie oznaczone jako L_n^d. Wyniki zapisano do plików CSV (`Ln_d1.csv` i `Ln_d2.csv`), a następnie przetworzono w Pythonie w celu uzyskania:

- wykresów wszystkich powtórzeń z naniesioną średnią,
- wykresów ilorazów średniego zapełnienia do funkcji teoretycznych: $f_1(n) = \frac{\ln n}{\ln \ln n}$ $f_2(n) = \frac{\ln \ln n}{\ln 2}$

Wyniki

Surowe wyniki

Na wykresach dla $d = 1$ i $d = 2$ widać, że:

Dla $\boldsymbol{d = 1}$ maksymalne zapełnienie rośnie wraz z n w sposób zbliżony do $\frac{\ln n}{\ln \ln n}$. Dla $\boldsymbol{d = 2}$ maksymalne zapełnienie jest znacznie mniejsze i rośnie wolniej, zgodnie z teoretycznym przewidywaniem $\frac{\ln \ln n}{\ln 2}$. W przypadku $d = 2$ wyniki są silnie skoncentrowane wokół wartości średniej, co pokazuje skuteczny efekt **równoważenia** urn przy wyborze dwóch opcji.

Wykresy ilorazów

Iloraz $\frac{L_n^1}{\frac{\ln n}{\ln \ln n}}$ jest prawie stały dla dużych n, co potwierdza asymptotyczną poprawność teorii dla $d = 1$. Iloraz $\frac{L_n^2}{\frac{\ln \ln n}{\ln 2}}$ również utrzymuje się blisko stałej wartości dla dużych n, co jednoznacznie wskazuje, że strategia **Power of Two Choices** skutecznie zmniejsza maksymalne zapełnienie.

Wnioski

1. Zastosowanie strategii dwóch losowych wyborów znacząco zmniejsza maksymalną liczbę kul w pojedynczej urnie.
2. Wyniki eksperymentalne są zgodne z przewidywaniami teoretycznymi:
   - L_n^1 rośnie proporcjonalnie do $\frac{\ln n}{\ln \ln n}$,
   - L_n^2 rośnie proporcjonalnie do $\frac{\ln \ln n}{\ln 2}$.
3. Dla $d = 2$ obserwuje się mniejszą wariancję wyników — strategia równoważy obciążenie urn w sposób wyraźnie skuteczny.

Zadanie 2

Cel eksperymentu

Celem eksperymentu było zbadanie zachowania algorytmu sortowania przez wstawianie (**Insertion Sort**) dla różnych rozmiarów danych. Eksperyment miał na celu:

1. Pomiar liczby porównań elementów tablicy podczas sortowania.
2. Pomiar liczby przestawień elementów (ruchów) podczas sortowania.

3. Analizę zależności liczby porównań i przestawień od rozmiaru tablicy oraz porównanie ich z teoretycznymi funkcjami rosnącymi liniowo i kwadratowo.

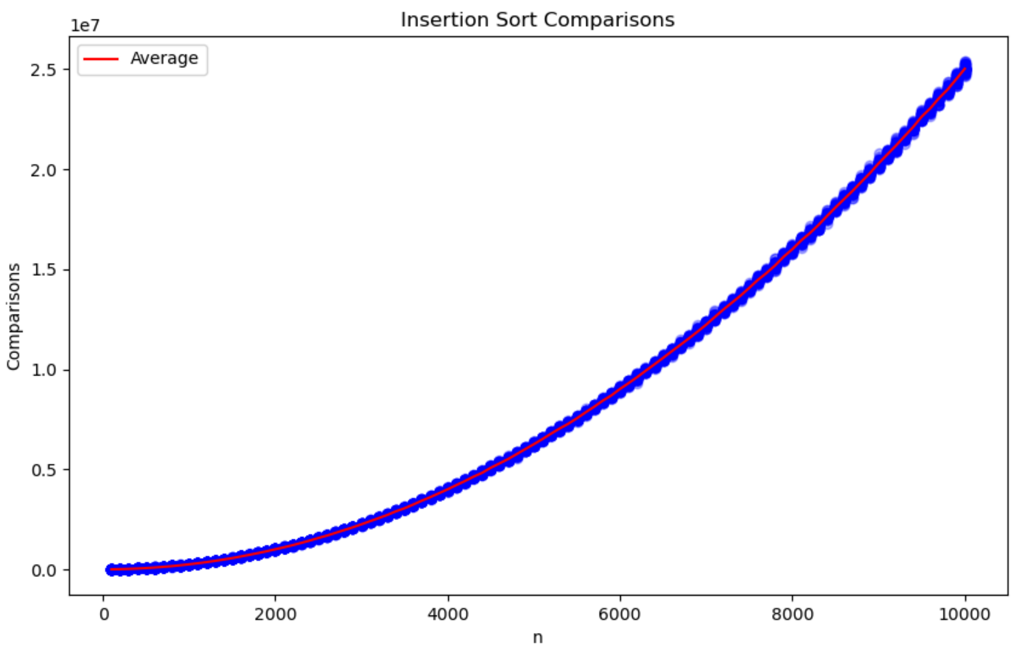
Metoda

Dla każdego $n \in \{100, 200, ..., 10000\}$ wykonano k = 50 niezależnych powtórzeń eksperymentu:

1. Generowano tablicę A[1..n] będącą losową permutacją liczb {1, ..., n} (każda permutacja była jednakowo prawdopodobna).
2. Sortowano tablicę algorytmem **Insertion Sort**, zliczając:
   - liczbę porównań elementów tablicy,
   - liczbę przestawień elementów (ruchów kluczy).
3. Wyniki zapisywano do plików CSV (`insertion_sort_results.csv`) w celu dalszej analizy.

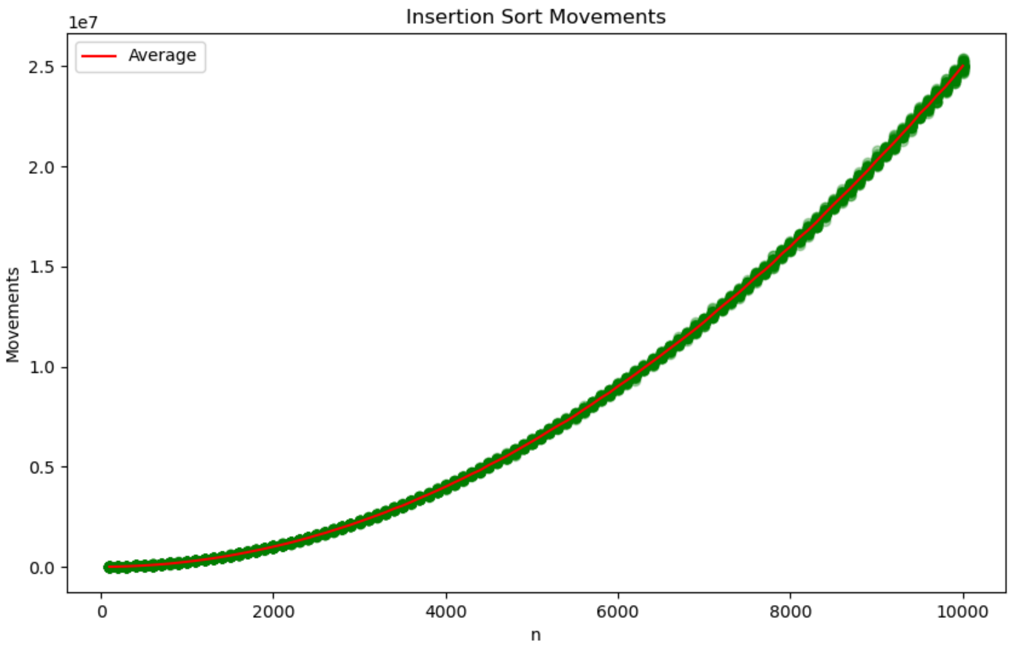
Wykresy

Liczba porównań



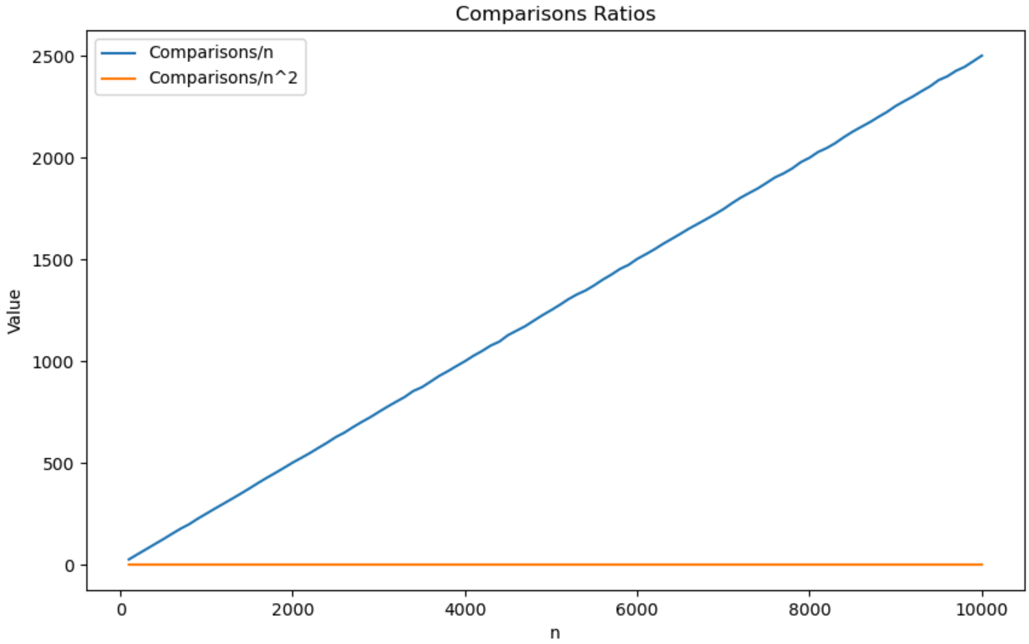
Na wykresie widać rosnącą liczbę porównań wraz ze wzrostem n. Średnia liczba porównań dla każdego rozmiaru n, oznaczona jako cmp(n), rośnie w przybliżeniu kwadratowo, zgodnie z teoretyczną złożonością $O(n^2)$ algorytmu **Insertion Sort**.

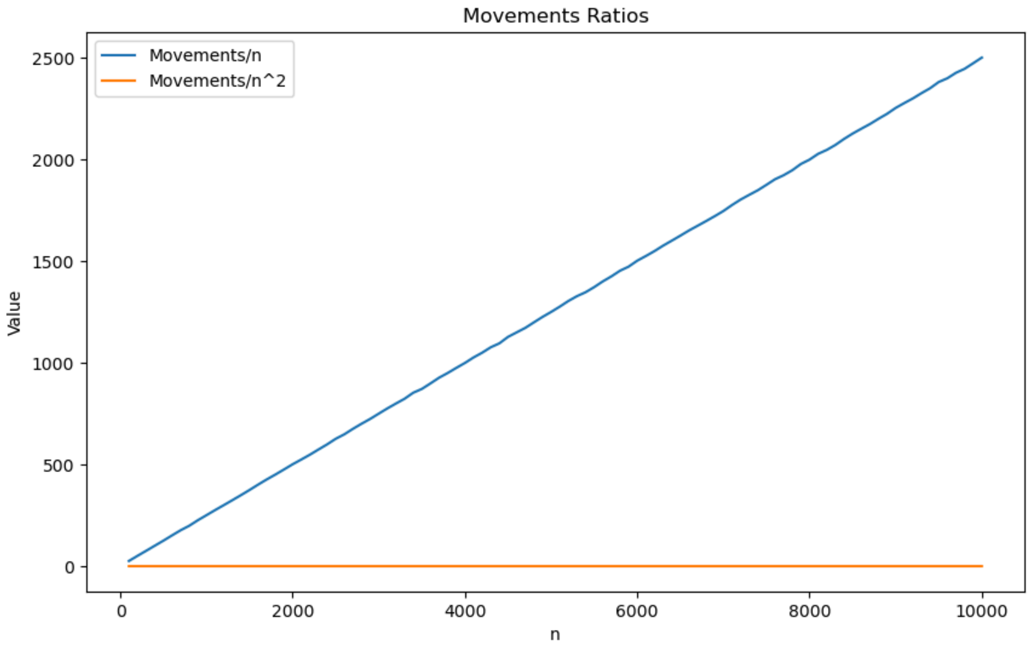
Liczba przestawień



Analogicznie, liczba przestawień s(n) również rośnie wraz z n i wykazuje charakterystyki zbliżone do kwadratowej zależności.

Ilorazy porównań i przestawień





Na wykresach przedstawiono:

- $cm\frac{p(n)}{n}$ oraz $cm\frac{p(n)}{n^2}$ jako funkcje n,
- $\frac{s(n)}{n}$ oraz $\frac{s(n)}{n^2}$ jako funkcje n.

Widać, że:

- iloraz $cm\frac{p(n)}{n}$ rośnie wraz z n, ale $cm\frac{p(n)}{n^2}$ zbliża się do stałej wartości dla dużych n, co potwierdza kwadratową złożoność czasową,
- analogicznie dla przestawień s(n).

Wnioski

1. Algorytm **Insertion Sort** wykazuje rosnącą liczbę porównań i przestawień w miarę zwiększania rozmiaru danych.
2. Średnia liczba porównań i przestawień rośnie w przybliżeniu jak $O(n^2)$, co jest zgodne z teoretycznymi oczekiwaniami.
3. Eksperyment potwierdza, że **Insertion Sort** jest efektywny dla małych tablic, ale jego wydajność szybko spada przy dużych danych.

Zadanie 3

Cel eksperymentu

Celem eksperymentu było zbadanie minimalnej liczby rund T_n, potrzebnej do rozesłania informacji z centralnej stacji (węzeł 0) do wszystkich pozostałych węzłów w sieci o topologii gwiazdy, przy uwzględnieniu zakłóceń. Eksperyment pozwalał oszacować, jak prawdopodobieństwo prawidłowego odbioru wiadomości przez pojedynczy węzeł wpływa na czas rozprzestrzeniania informacji.

Metoda

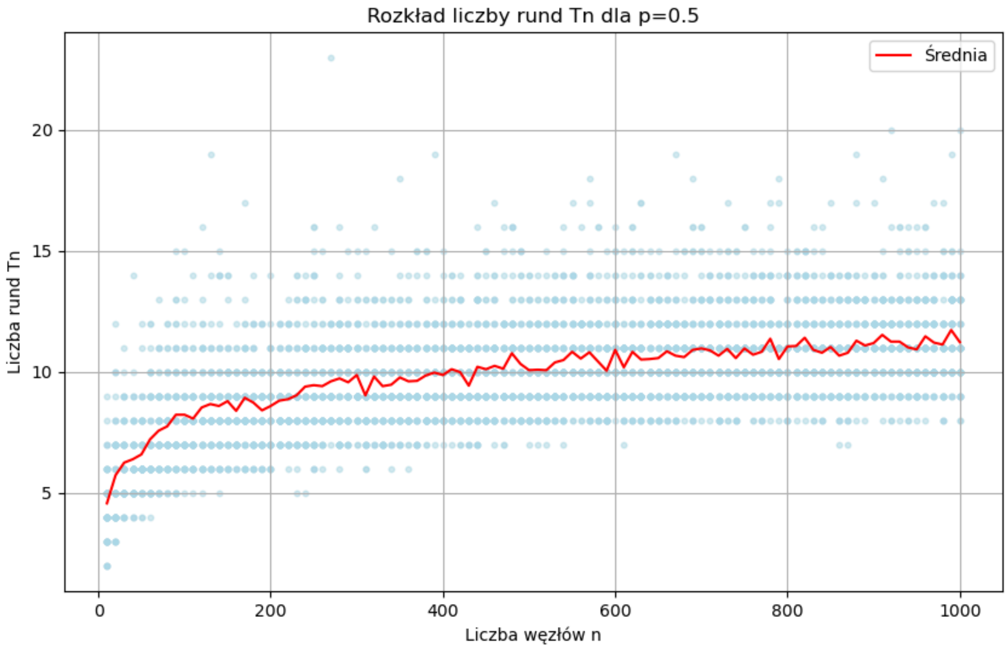
Dla każdego $n \in \{10, 20, ..., 1000\}$ wykonano k = 50 niezależnych powtórzeń eksperymentu dla dwóch wartości prawdopodobieństwa odbioru p = 0.5 oraz p = 0.1:

1. Węzły 1..n początkowo nie posiadają informacji.
2. W każdej rundzie stacja 0 wysyła wiadomość do wszystkich węzłów. Każdy węzeł otrzymuje wiadomość niezależnie z prawdopodobieństwem p.

3. Rundy powtarzano, aż wszystkie węzły otrzymały informację. Liczbę rund zapisano jako Tn.
4. Wyniki zapisano do plików CSV i wykorzystano w Pythonie do wygenerowania wykresów.

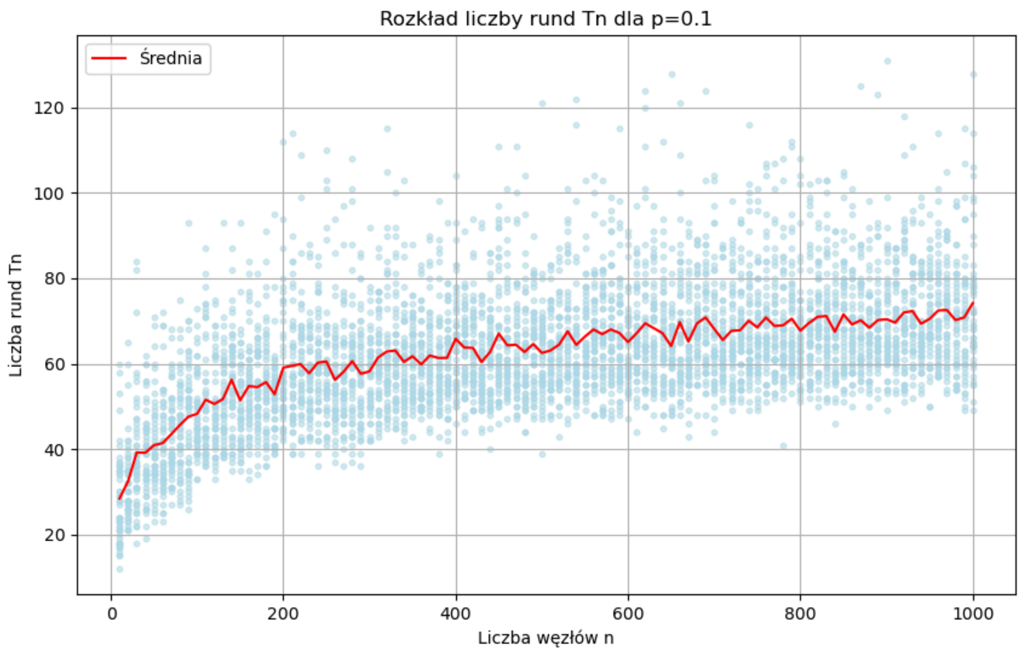
Wykresy

p = 0.5



Na wykresie widać rozrzut liczby rund w poszczególnych powtórzeniach (kolor niebieski) oraz średnią t(n) (kolor czerwony). Dla p = 0.5 średnia liczba rund rośnie powoli wraz z n, co oznacza, że komunikacja jest stosunkowo efektywna.

p = 0.1



Dla p = 0.1 rozrzut liczby rund jest znacznie większy, a średnia t(n) rośnie szybciej wraz z n. Oznacza to, że przy niskim prawdopodobieństwie odbioru zakłócenia znacznie wydłużają czas rozprzestrzeniania informacji.

Wnioski

1. Minimalna liczba rund Tn rośnie wraz ze zwiększaniem się liczby węzłów n.
2. Wyższe prawdopodobieństwo odbioru (p = 0.5) prowadzi do mniejszej liczby rund i mniejszego rozrzutu wyników.
3. Przy niskim prawdopodobieństwie odbioru (p = 0.1) liczba rund jest większa i bardziej zróżnicowana, co pokazuje znaczenie zakłóceń w procesie komunikacji.