Grzegorz Szulik 7 7 7 7

..... ----------

(Imię i nazwisko) (A, B, C, D)

Parametry:
M = 7
N = 77

Przykład raportu z Pracowni nr 1

Zadanie 1.

1. Cel zadania

Celem zadania było zbadanie złożoności obliczeniowej algorytmów sortowania przez wstawianie i przez wybieranie.

2. Metody

W doświadczeniu wykorzystano kilka klas stworzonych w języku Java. Odpowiedni projekt stworzono i kompilowano w środowisku NetBeans IDE 7.3 na komputerze przenośnym o procesorze Intel Celeron CPU 1007U.

3. Przebieg doświadczenia i wyniki

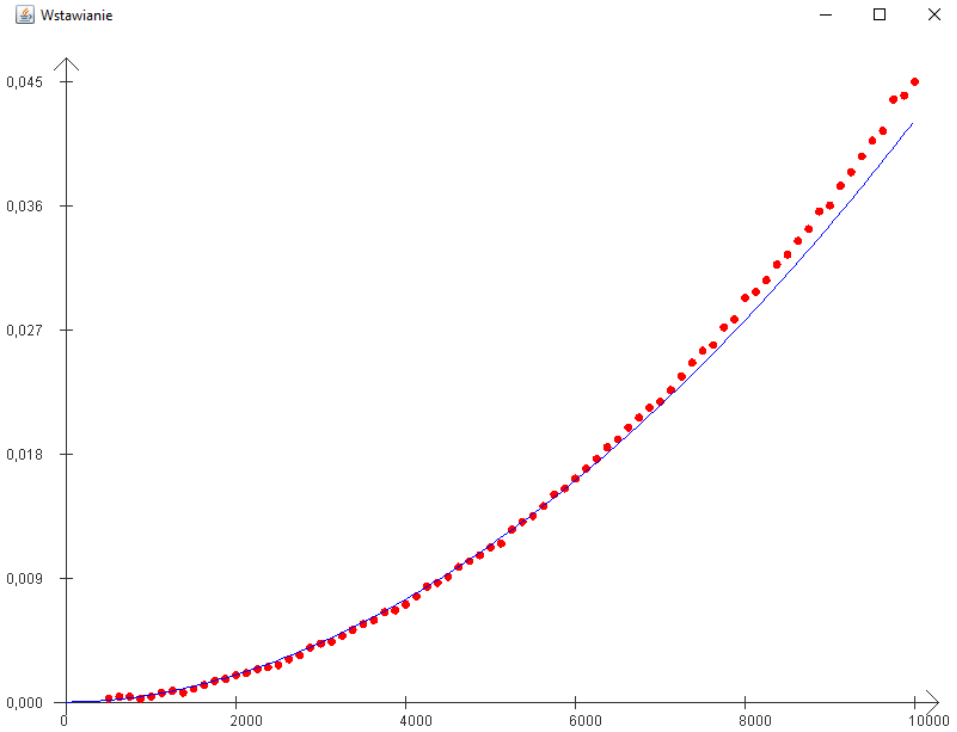
Doświadczenie rozpoczęto od ustalenia minimalnego i maksymalnego rozmiaru listy. Przyjęto, że będą to:

- *n_min* = 500, dla której czasy sortowania wyniosły powyżej 0,002 sekundy,
- *n_max* = 10 000, dla której czas sortowania wyniósł ok. 0,5 sekundy.

Opracowano metodę *MierzCzas*, która pozwala obliczyć czas sortowania się listy o zadanej długości jedną z dwóch badanych metod. Poniżej zamieszczono kod tej metody:

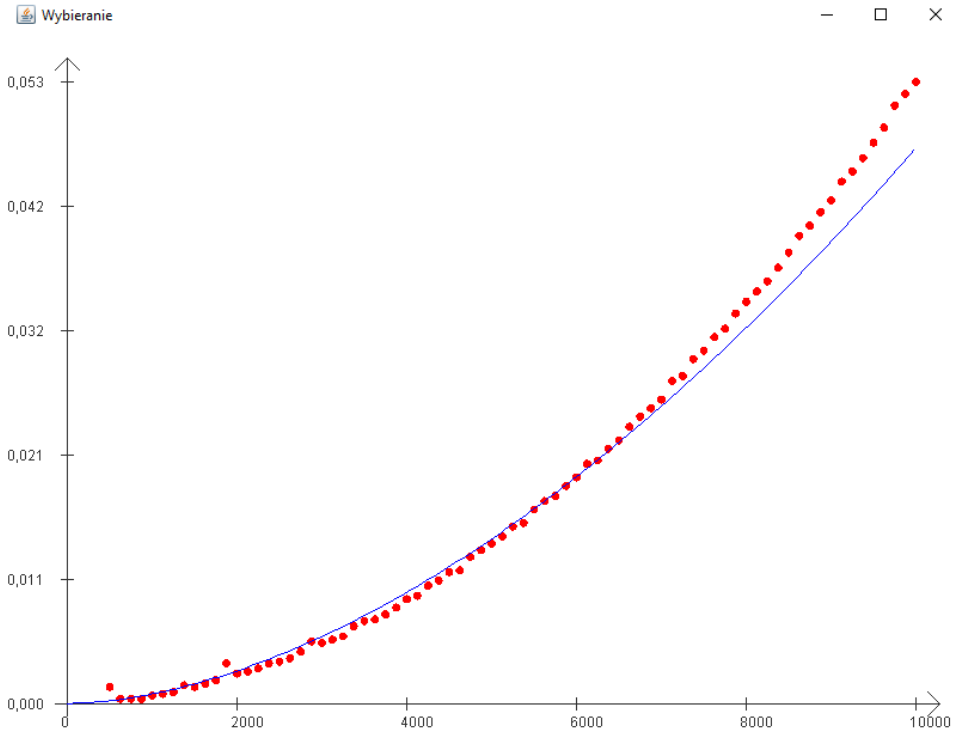
```
public double MierzCzas(int n, int metoda){
        double czas = 0.0;
        long pomiar;

        //mierze czas wykonywania sie M sortowan
        for(int i = 0; i < M; i++){
               this.Losuj(n);
               pomiar = System.currentTimeMillis();
               switch(metoda){
               //wybieram metode, ktora bedzie wykorzystana
                    case 1:   SortujPrzezWstawianie(n);
                              break;
                    case 2:   SortujPrzezWybieranie(n);
                              break;
               }
               pomiar = System.currentTimeMillis() - pomiar;
            czas += pomiar;
        }
        return czas / (M * 1000.0);
    }
```

Następnie wywołano metodę *BadajZlozonosc* podając jako parametry klasy *Sortowania* wartości $M = 7$ oraz $N = 77$.



Wykres 1. Zależność czasu sortowania listy metodą wstawiania od wielkości listy

Empiryczna złożoność obliczeniowa wyniosła: $n^{1,879}$, co bliskie jest teoretycznej wartości n^2. Podobny eksperyment przeprowadzono dla metody sortowania przez wybieranie.



Wykres 2. Zależność czasu sortowania listy metodą wybierania od wielkości listy

Tym razem uzyskano złożoność obliczeniową rzędu $n^{1.745}$, co zaskoczyło w pierwszym momencie autora opracowania ze względu na brak czasu optymistycznego dla sortowania przez wybieranie (w przypadku sortowania przez wstawianie czas optymistyczny jest rzędu n). Udało się ustalić przyczynę takiego zachowania – powodem jest brak definicji odpowiedniej struktury w rozwiązaniu – sortowanie przeprowadzane jest w oparciu o zwykłą tablicę i w momencie wstawiania elementu do tablicy trzeba sporo elementów przepisać do następnego pola. Niemniej jednak można zauważyć, że średnie czasy wykonywania się sortowania metodą wstawiania są niższe niż te uzyskane dla metody sortowania przez wybieranie.

4. Wnioski

W wyniku przeprowadzonego eksperymentu udało się oszacować złożoność obliczeniową algorytmu sortowania przez wstawianie i przez wybieranie. Otrzymana eksperymentalna złożoność jest bliska teoretycznej wartości $O(n^2)$.

Zadanie 2.

1. Cel zadania

Celem zadania było porównanie dwóch metod sortowania list – metodą wstawiania oraz metodą wybierania.

2. Metody

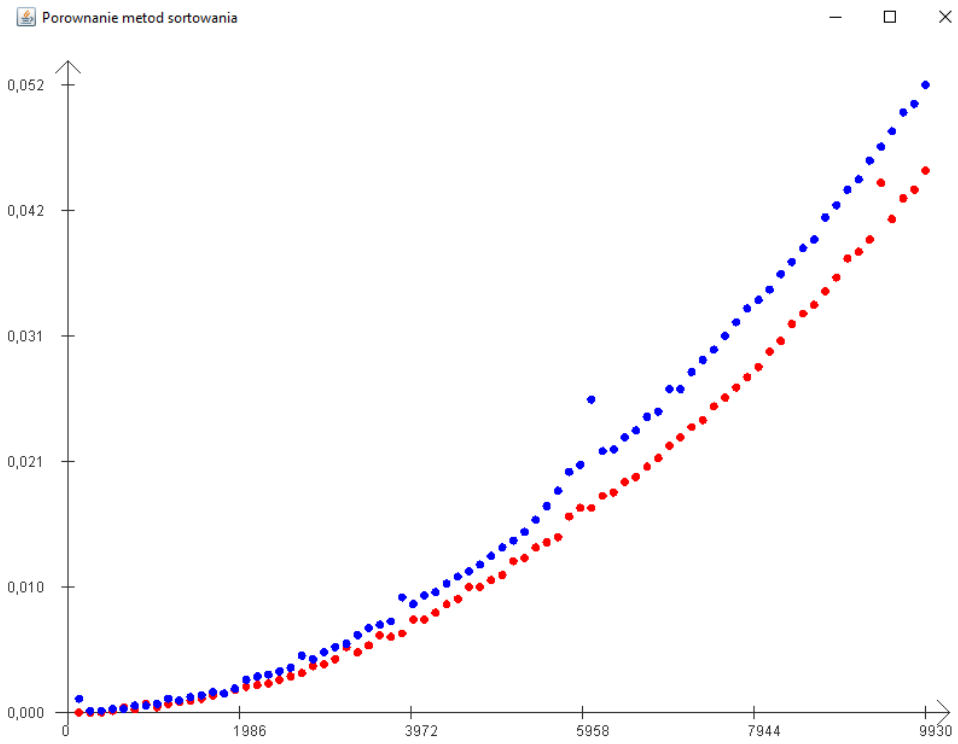
W doświadczeniu wykorzystano kilka klas stworzonych w języku Java. Odpowiedni projekt stworzono i kompilowano w środowisku NetBeans IDE 7.3 na komputerze przenośnym o procesorze Intel Celeron CPU 1007U.

3. Przebieg doświadczenia i wyniki

Wykorzystano maksymalną długość listy ustaloną w zadaniu 1. Ze względu na to, że w zadaniu 1 opracowano metodę *MierzCzas* w ten sposób, by umożliwiała zastosowanie obu metod sortowania można ją było wykorzystać w tym zadaniu. Po wywołaniu metody *PorownajMetody* dla obiektu klasy *Sortowanie* udało się uzyskać wykres zamieszczony na następnej stronie.

4. Wnioski

W wyniku przeprowadzonego doświadczania okazało się, że metoda sortowania przez wstawianie jest szybsza niż metoda sortowania przez wybieranie – różnice czasów dla krótkich list są niewielkie, ale rosną wraz ze wzrostem długości listy.



Wykres 3. Porównanie metod sortowania
(czerwone punkty – przez wstawianie, niebieskie – przez wybieranie)