

Algorytmy i struktury danych I

Proste algorytmy do sortowania

1 Bubble sort

Mając dany na wejściu ciąg liczb posortuj je rosnąco przy pomocy algorytmu bubble-sort.

Zastosuj kod programu generatorTest.cpp do wygenerowania serii liczb wejściowych. Za pomocą kodu wzorcowka.cpp wyprodukuj wzorcowy output. Porównaj poprawność outputu generowanego przez Twój generator.

1.1 Wejście

Pierwsza linia wejścia zawiera liczbę całkowitą z oznaczającą liczbę zestawów danych, których opisy występują kolejno po sobie. Opis jednego zestawu danych jest następujący:

W pierwszym wierszu zestawu danych znajduje się liczba n ($1 \leq n \leq 100000$).

W kolejnej linii znajduje się n liczb, które to należy posortować.

1.2 Wyjście

Dla każdego zestawu danych wypisz posortowane liczby.

1.3 Przykład

Dla danych wejściowych:

```
3
5
5 4 3 2 2
3
1 2 3
6
1 1 1 1 1 1
```

Poprawną odpowiedzią jest:

```
2 2 3 4 5
1 2 3
1 1 1 1 1 1
```

2 Inne algorytmy

Zrealizuj to samo zadanie dla następujących algorytmów:

- sortowanie przez wstawianie
- sortowanie przez zliczanie
- sortowanie przez scalanie

Jaka jest złożoność obliczeniowa tych algorytmów?

3 Zaliczenie zestawu

- Umieszczenie w systemie PEGAZ:
 - 1 plik z kodem programu: program main sterujący procedurą wczytywania i wypisywania danych, oraz wywołujący 4 funkcje sortująca, można wykorzystać kod programu *wzorcowka.cpp*.
 - 1 plik inputowy wykorzystany do testowania programów, w formacie jak opisany w tekście zadania, wygenerowany przy pomocy kodu *generatorTest.cpp*.
 - 1 plik outputowy w wyniku działania algorytmów sortujących, w formacie jak opisany w tekście zadania.
- Ustna odpowiedź: prezentacja działającego kodu i umiejętność przedstawienia algorytmu sortującego.