

## Algorytmy i struktury danych I

### Proste algorytmy do sortowania

## 1 Sortowanie przez wstawianie

Mając dany na wejściu ciąg liczb posortuj je rosnąco przy pomocy algorytmu *sortowanie przez wstawianie*. Zastosuj kod programu `genTest.cpp` do wygenerowania serii liczb wejściowych. Za pomocą kodu `sortTest.cpp` utwórz wzorcowy output posortowanych serii, kod używa funkcji bibliotecznej `sort`. Porównaj poprawność outputu wypisanego przez twój algorytm sortujący z wzorcowym outputem.

### 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  $n$  ( $1 \leq n \leq 100000$ ) oznaczająca ilość danych w danym zestawie.
- W kolejnej linii znajduje się  $n$  liczb tego zestawu, 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 zliczanie (ang. counting sort)
- sortowanie pozycyjne (ang. radix sort)
- sortowanie bąbelkowe (ang. bubble-sort)

Jaka jest złożoność obliczeniowa tych algorytmów?

## 3 Zaliczenie zestawu 1

Obowiązującym językiem programowania jest C/C++ oraz system operacyjny Linux. Rozwiązanie należy umieścić w systemie PEGAZ w postaci jednego skompresowanego pliku zawierającego:

- 1 plik z kodem programu: program main sterujący z procedurą wczytywania i wypisywania danych, wywołujący 4 metody sortujące. Można wykorzystać kod programu `sortTest.cpp` jako program main, zaimplementować tylko metody sortujące.
- 1 plik inputowy wykorzystany do testowania programów, w formacie jak opisany w tekście zadania, wygenerowany przy pomocy kodu `genTest.cpp`.
- 1 plik outputowy w wyniku działania algorytmów sortujących, w formacie jak opisany w tekście zadania.
- Plik `makefile` oraz `README` z instrukcją w jaki sposób uruchamiamy 4 algorytmy sortujące
- Plik dokumentacji w formacie `.pdf` lub `ascii`: krótki opis zastosowanych algorytmów sortujących oraz ich złożoności obliczeniowej.