Programowanie i metody numeryczne

Zadania – seria 11.

Sortowanie i normalizacja danych.

Zadanie 1. `selectionsort` – Sortowanie przez wybieranie.

Napisz szablon funkcji

```
template <typename ForwardIt>
void SelectionSort(ForwardIt begin, ForwardIt end)
```

wykonującej sortowanie metodą przez wybieranie. Funkcja powinna sortować dane zawarte pomiędzy dwoma iteratorami jednokierunkowymi `begin` i `end`.

Możesz wykorzystać:

- funkcję `std::iter_swap` z pliku nagłówkowego `algorithm` lub funkcję `std::swap` – do zamiany dwóch wartości,
- funkcję `std::min_element` z pliku nagłówkowego `algorithm` – do znalezienia iteratora wskazującego najmniejszy element pomiędzy dwoma zadanymi iteratorami.

Korzystając z tego szablonu, napisz program `selectionsort`, który wczytuje ze standardowego wejścia ciąg liczb całkowitych, a następnie wypisuje te liczby na standardowe wyjście w kolejności rosnącej.

Zadanie 2. `insertionsort` – Sortowanie przez wstawianie.

Napisz szablon funkcji

```
template <typename RandomIt>
void InsertionSort(RandomIt begin, RandomIt end)
```

wykonującej sortowanie metodą przez wstawianie. Funkcja powinna sortować dane zawarte pomiędzy dwoma iteratorami dostępu bezpośredniego `begin` i `end`.

Możesz wykorzystać następujące funkcje z pliku nagłówkowego `algorithm`:

- `std::upper_bound`, zwracającą iterator wskazujący na pierwszy element leżący pomiędzy dwoma zadanymi iteratorami, którego wartość jest większa niż pewna zadana wartość,
- `std::rotate`, przesuwającą elementy leżące pomiędzy dwoma zadanymi iteratorami w lewo, o tyle, by jeden z tych elementów znalazł się na początku.

Korzystając z tego szablonu, napisz program `insertionsort`, który wczytuje ze standardowego wejścia ciąg liczb całkowitych, a następnie wypisuje te liczby na standardowe wyjście w kolejności rosnącej.

Zadanie 3. `mergesort` – Sortowanie przez scalanie.

Napisz szablon funkcji

```
template <typename RandomIt>
void MergeSort(RandomIt begin, RandomIt end)
```

wykonującej sortowanie metodą przez scalanie. Funkcja powinna sortować dane zawarte pomiędzy dwoma iteratorami dostępu bezpośredniego `begin` i `end`.

Możesz wykorzystać funkcję `std::inplace_merge` z pliku nagłówkowego `algorithm` do scalania posortowanych podciągów.

Korzystając z tego szablonu, napisz program `mergesort`, który wczytuje ze standardowego wejścia ciąg liczb całkowitych, a następnie wypisuje te liczby na standardowe wyjście w kolejności rosnącej.

* Zadanie 4. `quicksort` – Sortowanie szybkie.

Napisz szablon funkcji

```
template <typename RandomIt>
void QuickSort(RandomIt begin, RandomIt end)
```

wykonującej sortowanie metodą sortowania szybkiego. Funkcja powinna sortować dane zawarte pomiędzy dwoma iteratorami dostępu bezpośredniego `begin` i `end`.

Możesz wykorzystać funkcję `std::partition` z pliku nagłówkowego `algorithm`, która dzieli elementy leżące pomiędzy dwoma zadanymi iteratorami na dwie grupy: w pierwszej znajdują się elementy spełniające określony warunek, zaś w drugiej – pozostałe.

Korzystając z tego szablonu, napisz program `quicksort`, który wczytuje ze standardowego wejścia ciąg liczb całkowitych, a następnie wypisuje te liczby na standardowe wyjście w kolejności rosnącej.

* Zadanie 5. `normalize` – Normalizacja danych.

Napisz szablon funkcji

```
template <typename ForwardIt, typename OutIt, typename T>
void Normalize(ForwardIt begin, ForwardIt end, OutIt out, T min, T max)
```

wykonującej normalizację danych. Funkcja powinna poddawać dane zawarte pomiędzy dwoma iteratorami jednokierunkowumi `begin` i `end` transformacji liniowej, w wyniku której największa spośród ich wartości będzie równa wartości zmiennej `max`, zaś najmniejsza spośród ich wartości – wartości zmiennej `min`. Otrzymane w ten sposób znormalizowane dane powinny być przekazywane do iteratora danych wyjściowych `out`.

Możesz wykorzystać następujące funkcje z pliku nagłówkowego `algorithm`:

- `std::minmax_element`, zwracającą parę iteratorów wskazujących na najmniejszy oraz największy spośród elementów leżących pomiędzy dwoma zadanymi iteratorami,
- `std::transform`, wykonującą zadaną transformację na wszystkich elementach leżących pomiędzy dwoma zadanymi iteratorami.

Korzystając z tego szablonu, napisz program `normalize`, który wczytuje ze standardowego wejścia ciąg liczb rzeczywistych oraz żądne wartości maksymalną i minimalną, a następnie wypisuje na standardowe znormalizowane liczby.

Opracowanie: Bartłomiej Zglinicki.