

ALGORYTMY I STRUKTURY DANYCH

SORTOWANIE

1 Stabilne scalanie w miejscu i sortowanie przez scalanie

Zadanie:

Zaprogramuj na trzy sposoby procedurę, która będzie scalała dwa uporządkowane i sąsiadujące ze sobą fragmenty tablicy (o długościach odpowiednio m i n):

- zwykle scalanie z wykorzystaniem zewnętrznego bufora (czas $O(m + n)$ i dodatkowa pamięć $\min(m, n) + O(1)$);
- scalanie w miejscu metodą Kronroda (czas $O(m + n)$ i dodatkowa pamięć $O(1)$);
- stabilne scalanie w miejscu oparte na metodzie redukcji — zadanie to sprowadza się do znalezienia mediany dwóch posortowanych ciągów (czas $O((m + n) \log(m + n))$ i dodatkowa pamięć $O(1)$).

Następnie zaprogramuj iteracyjną wersję *sortowania poprzez scalanie* z wykorzystaniem zaprogramowanej wcześniej procedury scalającej (w rezultacie dostaniesz trzy procesury sortowania poprzez scalanie odpowiadające różnym sposobom scalania).

Zadanie polega na zaaplikowaniu do takich samych danych każdej z trzech procedur sortujących (sprawdź czy wyniki się zgadzają), dokonania pomiarów ich czasów działania i rozstrzygnięciu która z nich jest najszybsza.

Uwaga:

Powinieneś mieć trzy osobne programy i uruchamiać je oddzielnie. Zmierz tylko czas działania samego sortowania (nie licz czasu wczytywania danych i wypisywania wyników) i wypisz go na standardowe wyjście dla błędów.

Dane:

W pierwszym wierszu z danymi jest podana liczba n ($1 \leq n \leq 1000000$) oznaczająca rozmiar danych, a w kolejnych n wierszach zapisane są liczby całkowite z zakresu od 0 do 1000000.

Wyniki:

W wyniku należy wypisać wszystkie spośród n liczb w kolejności niemalejącej (każdą w osobnym wierszu).

Sprawozdanie:

W sprawozdaniu należy zamieścić opis algorytmów i przeprowadzonych testów. Podziel testy na dwie grupy ze względu na wartości danych wejściowych: (i) każda liczba jest unikatowa, (ii) liczby mogą się powtarzać. Napisz, jakie czasy otrzymałeś/otrzymałaś testując swoje algorytmy w obu tych przypadkach.

2 Wybór mediany i sortowanie szybkie

Zadanie:

Zaprogramuj na dwa sposoby procedurę, która będzie wybierała któryś co do wielkości element spośród nieuporządkowanych danych zapisanych w określonym fragmencie tablicy (o długości n):

- wybór k -tego elementu zrandomizowanym algorytmem Hoare'a (czas oczekiwany to $O(n)$);
- wybór k -tego elementu deterministycznym algorytmem magicznych piątek Bloom–Floyda–Pratta–Rivesta–Tarjana (czas pesymistyczny to $O(n)$).

Następnie zaprogramuj rekurencyjną wersję *sortowania szybkiego* z podziałem według mediany z wykorzystaniem zaprogramowanej wcześniej procedury wybierającej (w rezultacie dostaniesz dwie procedury sortowania szybkiego odpowiadające różnym sposobom wyboru mediany). Dodatkowo zaprogramuj jeszcze dwie wersje *sortowania szybkiego*, w których wyboru pivotu dokonuje się nie poprzez znalezienie mediany, ale losowo wybiera się go spośród danych:

- wybór jednego pivotu i podział na dwie części;
- wybór pary pivotów i trójpodział.

Zadanie polega na zaaplikowaniu do takich samych danych każdej z czterech procedur sortujących (sprawdź czy wyniki się zgadzają), dokonania pomiarów ich czasów działania i rozstrzygnięciu która z nich jest najszybsza.

Uwaga:

Powinieneś mieć cztery osobne programy i uruchamiać je oddzielnie. Zmierz tylko czas działania samego sortowania (nie licz czasu wczytywania danych i wypisywania wyników) i wypisz go na standardowe wyjście dla błędów.

Dane:

W pierwszym wierszu z danymi jest podana liczba n ($1 \leq n \leq 1000000$) oznaczająca rozmiar danych, a w kolejnych n wierszach zapisane są liczby całkowite z zakresu od 0 do 1000000.

Wyniki:

W wyniku należy wypisać wszystkie spośród n liczb w kolejności niemalejącej (każdą w osobnym wierszu).

Sprawozdanie:

W sprawozdaniu należy zamieścić opis algorytmów i przeprowadzonych testów. Podziel testy na dwie grupy ze względu na wartości danych wejściowych: (i) każda liczba jest unikatowa, (ii) liczby mogą się powtarzać. Napisz, jakie czasy otrzymałeś/otrzymałaś testując swoje algorytmy w obu tych przypadkach.