

Теоретичні відомості (Лекція_4 Піраміди.pdf)

Хід роботи:

Частина 1. Побудова піраміди. Пірамідальне сортування.

1. Створити нову бібліотеку `Heap` (файли `Heap.h`, `Heap.cpp`).
2. У бібліотеці `Heap`, створити функції `Parent(...)`, `Left(...)` та `Right(...)` для навігації по піраміді.
3. У бібліотеці `Heap`, згідно описаних в теоретичній частині алгоритмів, створити функції `BuildMaxHeap(...)` та `MaxHeapify(...)` для побудови та підтримки властивості незростаючої піраміди.
4. У цій же бібліотеці створити функції `BuildMinHeap(...)` та `MinHeapify(...)` для побудови та підтримки властивості неспадної піраміди. Реалізація цих функцій є дзеркально симетричною до функцій `BuildMaxHeap(...)` та `MaxHeapify(...)`.
5. У бібліотеці `Heap`, згідно описаного в теоретичній частині алгоритму, створити функцію `HeapSort(...)` для реалізації пірамідального сортування одномірного масиву даних. У функції `HeapSort(...)` передбачити можливість сортування за зростанням (виклик функції `BuildMaxHeap(...)`) чи за спаданням (виклик функції `BuildMinHeap(...)`).
6. Створити новий проект `Lab_3_1` та підключити до нього бібліотеку `Heap`. У функції `main()` проекту реалізувати меню для введення одномірного масиву даних, побудову на основі цих даних незростаючої чи неспадної піраміди, сортування введених даних за зростанням чи за спаданням, відображення результатів.
7. Відкомпілювати проект та продемонструвати його роботу для одномірного масиву даних, отриманого від викладача.

Частина 2. Черга з пріоритетами.

1. У бібліотеці `Heap`, згідно описаних в теоретичній частині алгоритмів, створити функції `HeapMax (...)`, `HeapExtractMax (...)`, `HeapIncreaseKey (...)` та `MaxHeapInsert (...)` для роботи з чергою з пріоритетами, побудованою на основі незростаючої піраміди. Ці функції повинні, відповідно, повертати значення максимального елементу черги, видаляти максимальний елемент, змінювати (збільшувати) значення елементу, додавати елемент у чергу. У функції `HeapIncreaseKey (...)` передбачити обробку помилки зменшення ключа.
2. У цій же бібліотеці створити функції `HeapMin (...)`, `HeapExtractMin (...)`, `HeapDecreaseKey (...)` та `MinHeapInsert (...)` для роботи з чергою з пріоритетами, побудованою на основі неспадної піраміди. Реалізація цих функцій є дзеркально симетричною до функцій `HeapMax (...)`, `HeapExtractMax (...)`, `HeapIncreaseKey (...)` та `MaxHeapInsert (...)`. У функції `HeapDecreaseKey (...)` передбачити обробку помилки збільшення ключа.
3. Створити новий проект [Lab_3_2](#) та підключити до нього бібліотеку `Heap`. У функції `main()` проекту реалізувати меню для введення одномірного масиву даних, побудову на основі цих даних незростаючої чи неспадної піраміди та виконання описаних у п.1 та п.2 операцій для черги з пріоритетами, а також відображення результатів.
4. Відкомпілювати проект та продемонструвати його роботу для одномірного масиву даних, отриманого від викладача.