Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматики та управління в технічних системах

**Лабораторна робота №4**

З дисципліни «Алгоритми та структури даних-2»

Тема: «Дослідження структури даних: бінарної купи (binary heap)»

Виконали: Перевірив:

студенти групи ІТ-03 Смолій В. В.

Митєв А. Ю.

Яремчук Д. В.

Шевчук Д. Д.

Дата здачі: 25.04.21

Захищено з балом \_\_\_\_\_\_\_\_

Київ 2021

**Тема**: «Дослідження структури даних: бінарної купи (binary heap)»

**Мета**: Закріпити знання з механізмів функціонування такої структури даних, як бінарна купа(binary heap), реалізувати на практиці відповідні користувацькі структури даних та дослідити ефективність їх практичної реалізації.

**Завдання:**

Визначити склад, методи та інструменти для реалізації інтерфейсу та відповідного класу об'єктів, визначити алгоритми функціонування складових компонентів, реалізувати відповідне програмне забезпечення.

1. Реалізувати наступні методи для операцій:

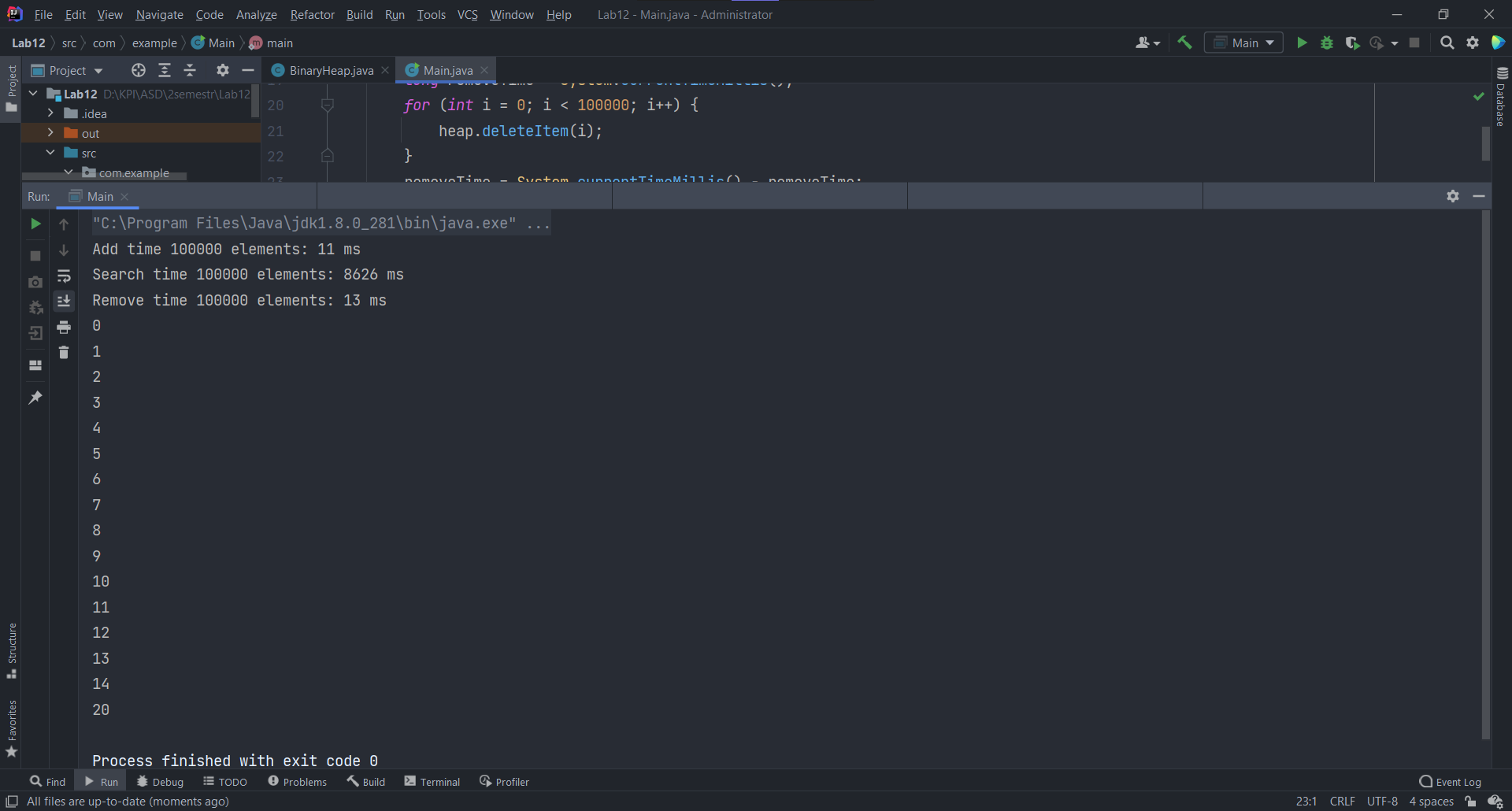
* вставки;
* видалення елементу;
* пошуку елементу за значенням (існуючого та не існуючого у структурі);
* отримання впорядкованої послідовності елементів.

2. Для кожної операції визначити найкращий та найгірший час виконання (на великому наборі числових значень, більше 10000) та пояснити отримані результати.

3. Визначити кількість пам’яті, яку займає структура даних.

**Хід роботи**

1. Вставка відбувається в два етапи: спочатку ми добавляємо необхідний елемент в кінець масиву, а потім запускаємо операцію heapify. Видалення: спочатку ми зменшуємо необхідний елемент до мінімуму, відбувається операція heapify, внаслідок якої цей елемент стає на першу позицію, потім ми видаляємо цей елемент масиву і знову виконуємо операцію heapify. Пошук по значенню відбувається за схожим принципом як і в звичайних деревах (з оглядкою на особливості цієї структури даних)
2. Вимірюємо час для 100000 об’єктів:

Як бачимо додавання та видалення елементів відбувається надзвичайно швидко. Найкращий час виконання буде, якщо елементи будуть додаватися в зростаючому порядку, найгірший – ж коли в спадаючому порядку. Але різниця між ними в часі буде не дуже суттєвою, адже в будь-якому випадку час цих двох операцій: для одного елемента – O(lg n), для n елементів – O(n\*lg(n)).

Як бачимо, пошук відбувається в цій структурі дуже повільно, адже ця структура не призначена для пошуку. В найгіршому випадку, ця операція займе O(n), а якщо ми шукаємо n елементів, то це взагалі займе O(n^2), тому краще не використовувати цю структуру даних для пошуку.

Також ми бачимо тут метод сортування. Так як ми використовуємо тут вбудований метод sort() для сортування, то швидкість буде – O(n\*log(n))

1. Розмір пам’яті:

Розмір масива – 4\*n + 4 (оскільки перший елемент масива ми не використовуємо)

Розмір змінної кількості елементів – 4 байти

Розмір змінної максимальної кількості елементів – 4 байти

Розмір структури даних – 4\*n + 4 + 4 + 4 = (4\*n + 12) байтів

1. Посилання на весь код - [https://github.com/Den-droid/ASD\_Lab](https://github.com/Den-droid/ASD_Lab3)4

***Висновки***

Ми ознайомилися з такою структурою даних як бінарна купа, дізналися її основні операції та час їхнього виконання.