

ЗАТВЕРДЖУЮ

Завідувач кафедри управління проєктами, інформаційних технологій та телекомунікацій, д.т.н., професор

Є. В. Мартин

« ____ » _____ 20 ____ р.

МЕТОДИЧНА РОЗРОБКА

для проведення практичного заняття з студентами (курсантами) 1-го курсу спеціальності 122 «Комп'ютерні науки» з дисципліни «Основи програмування та алгоритмізація»

Практичне заняття 5.4: «Застосування пірамід»

Мета: здобути навички щодо використання структури даних «піраміда» для задач сортування, а також виконання основних операцій з структурами даних: пошук, вставка та видалення елементів.

Час: 2 годин

Місце заняття: комп'ютерна лабораторія

Матеріальне забезпечення:

- ПК з відповідним ПЗ та доступом до мережі Internet;
- Методичні рекомендації щодо виконання практичної роботи.

Після практичного заняття студенти (курсанти) повинні:

вміти: реалізовувати алгоритм пірамідального сортування та писати ефективний код реалізації основних операцій у чергах з пріоритетами.

знати: особливості створення та відновлення властивостей пірамід, а також застосування пірамід для реалізації різних задач.

Література:

1. Віртуальне навчальне середовище ЛДУ БЖД «Віртуальний університет» [Електронний ресурс]. – Доступний з : <http://virt.ldubgd.edu.ua/course/view.php?id=1672>

План заняття:

1. Складання тестових завдань за темою попередньої лекції.
2. Реалізація алгоритму пірамідального сортування.
3. Видача завдань та виконання індивідуальних практичних завдань.

1. Складання тестових завдань за темою попередньої лекції.

Для складання тестових завдань необхідно перейти у відповідний розділ дисципліни «Основи програмування та алгоритмізація» у Віртуальному навчальному середовищі (ННІ цивільного захисту/спеціальність «Комп'ютерні науки»/бакалавр/І

курс/II семестр/Основи програмування та алгоритмізація/Тема 5. Структури даних/Контрольна частина/Тестове завдання за темою 5.3.)

Слід звернути увагу, що час та період складання тестового завдання обмежено.

2. Реалізація алгоритму пірамідального сортування.

Перед виконанням практичного завдання слід повторити лекцію «Піраміди», та нагадати особливості їх побудови та відновлення властивостей. Реалізацію алгоритму слід проводити з дотриманням таких вказівок: на вхід приймаємо масив з довільно наповненими елементами, далі а основі масиву будуємо піраміду, циклічно відновлюємо її властивість та повертаємо у вигляд відсортованого масиву.

Код реалізації алгоритму наведено нижче:

```
import java.util.Arrays;

public class HeapSort {

    public static void main(final String[] args) {
        int[] arr = new int[11];
        for (int i = 0; i < arr.length; i++) {
            arr[i] = ((int) (Math.random() * 12));
        }
        System.out.println(Arrays.toString(arr));

        heapSort(arr, arr.length);
        System.out.println(Arrays.toString(arr));
    }

    public static void heapSort(int[] array, int count) {
        heapify(array, count);

        int end = count - 1;
        while (end > 0) {
            swap(array, end, 0);
            siftDown(array, 0, --end);
        }
    }

    public static void swap(int[] array, int a, int b) {
        int temp = array[a];
        array[a] = array[b];
        array[b] = temp;
    }

    public static void heapify(int[] array, int count) {
        int start = count / 2 - 1;

        while (start >= 0) {
            siftDown(array, start, count - 1);
            start--;
        }
    }

    public static void siftDown(int[] array, int start, int end) {
```

```

    int root = start;

    while (root * 2 + 1 <= end) {
        int child = root * 2 + 1;
        int swap = root;
        if (array[swap] < array[child]) {
            swap = child;
        }
        if (child + 1 <= end && array[swap] < array[child + 1]) {
            swap = child + 1;
        }
        if (swap != root) {
            swap(array, root, swap);
            root = swap;
        } else {
            return;
        }
    }
}
}

```

3. Видача завдань та виконання індивідуальних практичних завдань.

В наведеному вище прикладі реалізації алгоритму пірамідального сортування представлені як операція побудови піраміди так і відновлення властивості піраміди. Ці операції потрібні для реалізації різноманітних дій у чергах з пріоритетами на основі пірамід

Отже, **умова завдання:** На основі програмного коду пірамідального сортування (використовуючи та модернізуючи його елементи) написати чотири методи у відповідності до представлених псевдокодів:

Вставки елементу:

MaxHeapInsert (A, key)

```

1 heap_size[A] ← heap_size[A] + 1
2 A[heap_size[A]] ←  $-\infty$ 
3 HeapIncreaseKey(A, heap_size[A], key)

```

Визначення максимального елемента в черзі:

HeapMaximum (A)

```

1 return A[1]

```

Видалення максимального елемента з черги:

HeapExtractMax (A)

```

1 if heap_size[A] < 1
2     then error "Черга порожня"
3 max ← A[1]
4 A[1] ← A[heap_size[A]]
5 heap_size[A] ← heap_size[A] - 1
6 MaxHeapify(A, 1)
7 return max

```

Збільшення ключа елементу:

```
HeapIncreaseKey(A, i, key)
1  if key < A[i]
2      then error "Новий ключ менше поточного"
3  A[i] ← key
4  while i > 1 та A[Parent(i)] < A[i]
5      do Обміняти A[i] ↔ A[Parent(i)]
6      i ← Parent(i)
```

Код написаної програми необхідно завантажити у віртуальне навчальне середовище до категорії «Застосування пірамід». Слід зауважити, що період виконання завдання обмежений.

Методичну розробку розробив:
Заступник начальника кафедри УПІТтаТ
майор служби цивільного захисту

О.В. Придатко

Методична розробка обговорена на засіданні кафедри УПІТтаТ
Протокол № ____ від "____" _____ 20__ р.