

ПИТАННЯ  
до екзамену

1. Дайте визначення структури даних і наведіть класифікацію, приклади.
2. Дайте визначення такої структури даних як «дерево», наведіть її класифікацію, приклади і способи переміщення по дереву.
3. Дайте визначення такої структури даних як «хеш-таблиця» і наведіть класифікацію, приклади.
4. Опишіть способи реалізації хеш-функції (перетворення ключа в індекс).
5. Опишіть способи вирішення колізії в хеш-таблицях з відкритою адресацією.
6. Дайте визначення алгоритму і опишіть його властивості.
7. Дайте поняття рекурсії, наведіть її особливості, приклад.
8. Дайте поняття терміну "сортування" і визначте групи алгоритмів сортування.
9. Опишіть сутність алгоритмів групи елементарного сортування (вибору, вставки, бульбашки, Шелла).
10. Опишіть сутність алгоритмів групи елементарного сортування (карманого, розподільного підрахунку).
11. Опишіть сутність алгоритмів групи «швидкого» сортування, наведіть їхню класифікацію і відмінності.
12. Опишіть сутність алгоритмів групи сортування злиттям, наведіть їхню класифікацію і особливості.
13. Дайте визначення такої структури даних як «черга за пріоритетом», наведіть її представлення і реалізацію.
14. Опишіть сутність базового алгоритму пірамідального сортування, наведіть відмінності у реалізації з чергою за пріоритетом.
15. Опишіть сутність групи алгоритмів порозрядного сортування, їх особливості, приклад.
16. Дайте визначення поняттю "пошуку", наведіть особливості реалізації і класифікацію алгоритмів пошуку.
17. Опишіть сутність послідовного, бінарного та інтерполяційного пошуку.
18. Дайте визначення BST-дерева і наведіть особливості реалізації пошуку на основі таких дерев.
19. Опишіть операції "ротація-вліво" і "ротація-вправо", наведіть їх призначення та приклад.
20. Розкрийте необхідність аналізу алгоритмів, наведіть методи аналізу.
21. Поясніть сутність емпіричного аналізу алгоритмів, його особливості й причини застосування.
22. Поясніть сутність математичного аналізу, його особливості й причини застосування.
23. Дайте поняття "рост-функція", призначення. Наведіть типи рост-функцій.

Три типа заданий:

1. Теоретическое задание (12 баллов)
2. Практическое задание (16 баллов)
3. Задача (12 баллов)

Оценки за задания		
В балах		Оценка по национальной шкале
Задание 2	Задание 1, 3	
14 - 16	11 - 12	Отлично
12 - 13	9 - 10	Хорошо
10 - 11	7 - 8	Удовлетворительно
меньше 10	меньше 7	Неудовлетворительно

Пример практического задания 1.

Дано описание класса **HashTable** (хэш-таблица с открытой адресацией):

```
class Obj {
    int data;
    short key;
    Obj(int a) {
        key = (short)(Math.random()*10);
        data = a;
    }
}
class HashTable {
    private Obj hash[];
    HashTable(int s) { hash = new Obj [s]; }
    // описание метода вывода
    void prin() {
        for(int i=0; i<hash.length; i++)
            if (hash[i] == null)
                System.out.println(i);
            else
                System.out.println(i + ": " + hash[i].data + "(" + hash[i].key + ")");
    }
    int Index(short k){
        return k%hash.length;
    }
}
```

Добавьте в класс **HashTable** описание метода вставки нового объекта типа **Obj** (предполагается, что размерность хэш-таблицы превышает или равна количеству хранимых объектов; при возникновении коллизии выполняется квадратичное зондирование).

Решение:

```
void ins(Obj a) {
    int ind = Index(a.key);
    int ind0 = ind;
```

```

for(int i = 0; i<hash.length; i++) {
    ind = (ind0 + (int)Math.pow(i, 2))%hash.length;
    if (hash[ind] == null) {
        hash[ind] = a;
        break;
    }
}
}
}

```

Пример практического задания 2.

Дано описание класса **Spisok**:

```

class Spisok {
    class Element {
        int data;
        Element next;
        Element(int x) { data=x; }
    }
    Element first;
    void prin() {
        Element cur = first;
        while (cur != null) {
            System.out.print(cur.data + "\t");
            cur = cur.next;
        }
    }
}

```

Добавьте в класс **Spisok** описание метода вставки нового элемента.

Решение:

```

public boolean ins(int data) {
    Element temp = new Element(data);
    if (first == null) first = temp;
    else
        if (first.data < temp.data) { temp.next = first;
                                      first = temp; }
        else
            first = insElem(first, temp);
    return true;
}
private Element insElem(Element cur, Element newEl) {
    if (cur.next != null)
        if ((cur.data >= newEl.data) && (cur.next.data < newEl.data)) {
            newEl.next = cur.next;
            cur.next = newEl;
        }
        else
            cur.next = insElem(cur.next, newEl);
    else
        cur.next = newEl;
    return cur;
}

```

Примеры задач.

1. Проанализировать работу алгоритма сортировки выборкой на нижеприведенном наборе данных и отобразить состояния этого набора данных на пяти итерациях выполнения алгоритма:

0	1	2	3	4	5	6	7	8	9
5	0	10	-3	4	7	-2	11	2	7

2. Проанализировать работу алгоритма бинарного поиска для значения 24 на нижеприведенном наборе данных и отобразить итерации необходимые для выполнения этого алгоритма:

0	1	2	3	4	5	6	7
-4	5	82	7	24	-115	12	104