

## Алгоритмы и структуры данных. Экзаменационные билеты.

1. Анализ вычислительной сложности алгоритмов. Нотация  $O(n)$ . Классификация алгоритмов по вычислительной сложности. Порядок роста вычислительной сложности. Примеры алгоритмов различной вычислительной сложности
2. Алгоритм Хаффмана (получения оптимальных префиксных кодов). Пример реализации. Сложность.
3. Задача о покрытии строки. Пример реализации. Сложность
4. Алгоритмы сортировки пузырьком и вставками. Особенности. Пример реализации. Сложность.
5. Алгоритм сортировки Шелла. Особенности. Пример реализации. Сложность.
6. Алгоритмы сортировки выбором и подсчетом. Особенности. Пример реализации. Сложность.
7. Алгоритм нахождения  $k$ -й статистики. Пример реализации. Сложность.
8. Алгоритм быстрой сортировки. Пример реализации. Сложность.
9. Алгоритм сортировки слиянием. Пример реализации. Сложность.
10. Алгоритм поразрядной сортировки. Пример реализации. Сложность.
11. Внешняя сортировка. Пример реализации. Сложность.
12. Линейная структура – стек. Пример реализации на основе обычного массива. Сложность основных операций.
13. Линейная структура – очередь. Пример реализации на основе обычного массива. Сложность основных операций.
14. Линейная структура – односвязный список. Пример реализации. Сложность основных операций.
15. Деревья. Классификация деревьев. Способы программного представления деревьев. Пример реализации.
16. Бинарная куча. Назначение. Добавление и удаление элементов. Пример реализации. Сложность основных операций.
17. Пирамидальная сортировка. Пример реализации. Сложность. Особенности.
18. Бинарное дерево отрезков. Пример использования. Сложность.
19. Бинарный поиск. Условия применимости. Пример реализации. Сложность.
20. Бинарное дерево поиска. Реализация в стандартных библиотеках. Сложность операций. Назначение балансировки.
21. Схемы обхода бинарных деревьев. Примеры использования различных схем обхода бинарных деревьев в разных задачах.
22. Задача о рюкзаке. Пример реализации и вычислительная сложность.
23. Вычисление расстояния редактирования. Пример реализации. Вычислительная сложность.
24. Пусть требуется найти произведение двух натуральных чисел  $a$  и  $b$ , используя лишь операции сложения и сравнения ( $>$ ,  $<$ ,  $>=$ ,  $<=$ ,  $!=$ ,  $==$ ). Очевидно, что это можно сделать за  $O(\min(a,b))$  сложений:  

```
long long mul1(int a, int b)
{
    long long i, sum=0;
    for(i = 0; i < b; i++) sum += a;
    return sum;
}
```

Приведите пример реализации алгоритма, получающего тот же результат за меньшее количество операций, и оцените его вычислительную сложность. Можно использовать циклы, условия, а также операции: присваивание, сложение и сравнений. Остальные операции, в том числе поразрядные, использовать нельзя.

25. Дана строка из заглавных букв латинского алфавита. Необходимо найти наибольший палиндром (и его длину), который можно получить вычеркиванием некоторых букв из данной строки. Палиндром – строка, которая одинаково читается слева направо и справа налево.

Пример входных данных:

***АВАССВА***

Пример выходных данных:

***АВССВА (6 символов)***

Описание эффективного алгоритма. Реализации в виде программы на одном из языков программирования: C++, C#, Java или Python по вашему выбору.

Определение вычислительной сложности и затрат памяти для этого алгоритма.

26. Задача о покрытии прямоугольника (размером 5x6). Пример реализации.

Вычислительная сложность.

27. Имеется массив из N точек на плоскости с целочисленными координатами.

Требуется найти периметр наименьшего охватывающего многоугольника, содержащего все точки.

Формат входного файла:

В первой строке N – число точек;

В каждой из следующих N строк два целых числа – координаты x и y очередной точки.

Формат выходных данных:

Одно вещественное число – периметр требуемого многоугольника с точностью до сотых (два знака после запятой).

Пример

Входной файл	Стандартный вывод
5 2 1 2 2 2 3 3 2 1 2	5.66

Описание эффективного алгоритма. Реализации в виде программы на одном из языков программирования: C++, C#, Java или Python по вашему выбору.

Определение вычислительной сложности и затрат памяти для этого алгоритма.