1. Оцените количество сравнений в функции void make\_heap(int \*a, int n); **o(n) в лучшем случае/ o(n logn) в остальных**
2. Оцените количество сравнений в функции int partition(int \*a, int n, int pivot\_pos); **o(n logn)/o(n\*n)**
3. Оцените минимальное количество обменов при быстрой сортировке.**omega(n logn)**
4. Определите количество сравнений при сортировке массива [1,2,3,4,5,6,7,8,9,0] по возрастанию выбором. **45**
5. Определите количество сравнений при сортировке массива [1,2,3,4,5,6,7,8,9,0] по возрастанию вставками. **18**
6. Напишите пример killer-sequence длиной 7 для быстрой сортировки с алгоритмом выбора опорного элемента: первый элемент. **(1 2 3 4 5 6 7)**
7. Напишите пример killer-sequence длиной 7 для быстрой сортировки с алгоритмом выбора опорного элемента: средний элемент. **(9 9 9 1 9 9 9)**
8. Напишите пример последовательности длиной 7, на котором достигается минимальное время работы quicksort с алгоритмом выбора пивота «первый элемент». **(4 1 2 3 5 6 7)**
9. Приведите пример 2х отсортированных последовательностей длины 10, для которых количество сравнений при слиянии максимально. **1 3 5 7 9 11 13 15 17 and 2 4 6 8 10 12 14 16 18 20**
10. Опишите алгоритм пирамидальной сортировки. **(1. Строим кучу на исходном массиве.**

**2. N − 1 раз достаем максимальный элемент,**

**кладем его на освободившееся место**

**в правой части.)**

1. Перечислите достоинства и недостатки пирамидальной сортировки. **(дос.: 1. Сортирует на месте требует О(1)**

**памяти;**

**Всегда работает за O(nlogn); недос.: 1. Нестабильна;**

**Почти отсортенные данные сортирует также долго, как хаотичные)**

1. Опишите алгоритм quicksort. **(1. Выбор элемента 2. Перераспределение: меньшие эл-ты вперед, большие назад 3. Рекурсивно применить п1 и п2 к двум подмассивам)**
2. Перечислите достоинства и недостатки quicksort. **(дос.: Сортирует на месте требует О(1)**

**памяти; лучшая и средняя работа за O(nlogn);**

**недос.: 1. Нестабильна;**

**в худшем случае работает за n\*n)**

1. В чём смысл оптимизации концевой рекурсии? **(хвост рекурсия (где последняя операция это вызов рекурсии) преобразуется компилятором в цикл, тк обычная рекурсия при большом количестве посылок может произойти переполнение стека)**
2. Опишите алгоритм сортировки слиянием. **(1. Разбить массив на два.**

**2. Отсортировать каждый (рекурсивно).**

**Слить отсортированные в один.)**

1. В чём разница между методами восходящей и нисходящей сортировки слиянием **(в восходящем методе отсутствует рекурсивное деление массива на пары/тройки)**
2. Предложите алгоритм слияния нескольких отсортированных массивов (k > 2) **(Построим min-heap из первых элементов A1 0 , A2 0 , ... , Ak 0 .**

**Пока куча не пуста: – Скопируем минимум из кучи в результат.**

**– Если минимальный элемент – последний в своем массиве, то извлечем элемент из кучи, иначе заменим его следующим из того же массива.**

**– Восстановим свойство кучи – SiftDown(0).**

**Чтобы определять индекс массива по элементу в куче, будем хранить в куче не сам элемент, а пару <индекс массива, индекс элемента>**

1. Какова сложность алгоритма нахождения k-й порядковой статистики с алгоритмом выбора опорного элемента “медиана трёх” в худшем случае? **(n^2)**
2. Какова сложность алгоритма нахождения k-й порядковой статистики с алгоритмом выбора опорного элемента “медиана медиан” в худшем случае? **(n)**
3. Какова сложность алгоритма нахождения k-й порядковой статистики с алгоритмом выбора опорного элемента “случайный элемент” в худшем случае? **(n^2)**
4. Какое минимальное количество сравнений достаточно для сортировки 5 чисел? **10**
5. Можно ли упорядочить 10 произвольных чисел, проведя 12 сравнений? **нет**
6. Поясните свойство стабильности алгоритма сортировки.**(сохранение следования порядка одинаковых элементов)**
7. Поясните свойство локальности алгоритма сортировки.**(не требует доп памяти сортирует на месте)**
8. Сколько двухпутевых слияний достаточно, чтобы отсортировать массив из 147 элементов? **28**
9. Как реализовать алгоритм сортировки, работающий не медленнее quicksort в среднем и с ассимптотикой O(n\*log(n)) в худшем случае? **Для этого надо сделать подобно сортировке Introsort. Нужно сортировать массив при помощи quicksort и считать глубину рекурсии. При превышении определенной глубины, например, логарифма из числа элементов, нужно переключиться на пирамидальную сортировку. Выбор пал в ее сторону, потому что она локальная и стабильная. Т.о. в худшем случае, когда quicksort работал бы n^2 мы все равно получим n\*logn. HeapSort, Mergesort**
10. Предложите алгоритм частичной сортировки первых K элементов из массива размера N с помощью слияния?

(**Так как mergeSort обычно определен рекурсивно, то достаточно вызвать его на требуемом участке массива. Если mergeSort определен как mergeSort(T\* arr, int count), и массив T A[n], то достаточно вызвать mergeSort(A, k))**

1. Предложите алгоритм частичной сортировки первых K элементов из массива размера N с помощью кучи? **(Нужно просто запустить heapSort на первых K элементов, то есть построить кучу только на этих элементах и перемещать требуемые элементы в конец кучи, который расположен не дальше K.)**
2. Оцените затраты времени и памяти в алгоритме сортировки подсчётом. **T(n,k)=M(n,k)=O(n+k)**
3. Опишите алгоритм сортировки подсчётом. **(Заведем массив C[0..k-1], посчитаем в C[i] количество вхождений**

**элемента i в массиве A.**

**• Выведем все элементы C по C[i] раз.)**

1. Опишите алгоритм поразрядной сортировки от младших разрядов к старшим.**(** **Сначала сортируем подсчетом по младшим разрядам, затем по старшим.**

**Ключи с различными младшими разрядами, но одинаковыми старшими**

**не будут перемешаны при сортировки старших разрядов, благодаря**

**стабильности поразрядной сортировки.)**

1. Опишите алгоритм поразрядной сортировки от старших разрядов к младшим. **(Сначала сортируем подсчетом по старшим разрядам, затем по младшим.**

**Чтобы не перемешать отсортированные старшие разряды, сортируем по**

**младшим только группы чисел с одинаковыми старшими разрядами**

**отдельно друг от друга.)**

1. В чём отличие алгоритмов binary quicksort и quicksort c выбором опорного элемента “середина”?**(1. Сравнение с мнимым элементом в бинсорт 2. Бинсорт запускает деление и сравнение каждый раз от нового разряда начиная с максимального 3. Надо найти опорный элемент (100…0 нули-к-во разрядов в максимальном числе))**
2. Какой метод лучше всего подходит для сортировки миллиона 6-значных десятичных чисел? **(quicksort, heapsort)**
3. Какой метод лучше всего подходит для сортировки строковых ключей разной длины? Почему? **(LSD сортируeт поразрядно, начиная с младшего)**