***Лабораторная работа 7.   
Алгоритмические стратегии***

1. Разработать программу поиска решения задачи о «магическом квадрате» третьего порядка методом полного перебора.
2. Преобразуй и властвуй. Даны n целых чисел и ряд из n коробок. Между коробками фиксировано расположены знаки неравенств. Разработайте программу, которая «помещает» числа в коробки так, чтобы неравенства были правильные.

Пример. Числа 2, 5, 1, 0. Знаки: < > <. Решение: 0<5>1<2.

1. Задача интервального разбиения. В задаче интервального планирования используется один ресурс и много заявок в форме временных интервалов; требуется выбрать, какие заявки следует принять, а какие отклонить. В задаче интервального разбиения используется несколько идентичных ресурсов, для которых необходимо распланировать все заявки с использованием минимально возможного количества ресурсов. Найдите жадный алгоритм решения и реализуйте его.

Порядковые статистики

1. Разработать программную реализацию алгоритма поиска k-ой статистики за линейное в среднем время.
2. Разработать программную реализацию алгоритма поиска k-ой статистики за линейное в худшем случае время.
3. Поиск k-ой порядковой статистики в двух массивах. Даны два отсортированных массива a и b размерами n и m соответственно. Требуется найти k-ый порядковый элемент   после их слияния. Считается, что все элементы в массивах различны и нумеруются с нуля.

*Простое решение.* Сольем два массива и просто возьмем элемент с индексом k-1. Слияние будет выполнено за время O(n+m), и использует O(n+m) дополнительной памяти.

*Чуть менее простое.* Используются два указателя, с помощью которых сможем обойти массивы, не сливая их. Первоначальное значение указателей – на начало каждого из массивов. Будем увеличивать на единицу тот из них, который указывает на меньший элемент. После (k-1)-ой итерации сравним элементы, на которых стоят указатели. Меньший из них и будет ответом. Таким образом, мы получим k-ый элемент за O(k) шагов.

*Хорошее решение.* В первом массиве выберем элемент c индексом i=n/2 и бинарным поиском найдем во втором массиве позицию j, на которой стоит наибольший элемент, меньший a[i]. Если i+j=k-2, то мы нашли k-ую порядковую статистику — это элемент a[i]. Иначе, если i+j>k-2, то далее тем же способом ищем в массиве a в диапазоне индексов [0,i-1], а если i+j<k-2, то в диапазоне индексов [i+1,n-1]. Решая задачу таким способом, мы получим асимптотику O(log(n)∙log(m)).

*Реализовать все три варианта и сравнить время работы программ.*

Баллы: задачи 1–5 оцениваются по одному баллу, задача 6 оценивается в 2 балла.