

# ЖАДНЫЕ АЛГОРИТМЫ

+ o

Школа::Кода  
Олимпиадное  
программирование

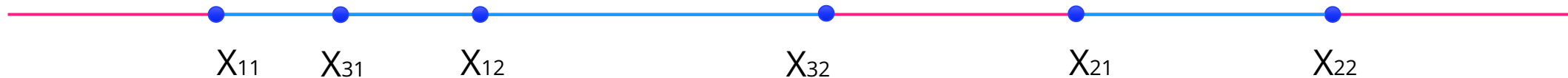
2020-2021 Таганрог

# Что такое жадные алгоритмы?

Жадный алгоритм — алгоритм, заключающийся в принятии локально оптимальных решений на каждом этапе, допуская, что конечное решение также окажется оптимальным.

# Задача об отрезках

Даны  $N$  отрезков на прямой, т.е. каждый отрезок задаётся парой координат  $(X1, X2)$ . Рассмотрим объединение этих отрезков и найдём его длину.



# Решение

Положим все координаты концов отрезков в массив  $X$  и отсортируем его по значению координаты.

Дополнительное условие при сортировке - при равенстве координат первыми должны идти левые концы. Кроме того, для каждого элемента массива будем хранить, относится он к левому или к правому концу отрезка. Теперь пройдемся по всему массиву, имея счётчик  $C$  перекрывающихся отрезков. Если  $C$  отлично от нуля, то к результату добавляем разницу  $X_i - X_{i-1}$ . Если текущий элемент относится к левому концу, то увеличиваем счётчик  $C$ , иначе уменьшаем его.

# Реализация

```
for (auto item : a)
{
    points.push_back(make_pair(item.first, -1));
    points.push_back(make_pair(item.second, 1));
}
sort(points.begin(), points.end());
int res = 0;
int count = 0;
int prev = points[0].first;
for (auto point : points)
{
    if (count)
    {
        res += point.first - prev;
    }
    count -= point.second;
    prev = point.first;
}
```

# Задача

Дан массив из  $N$  положительных чисел, надо найти в нем несколько чисел, идущих подряд, так, чтобы их сумма была больше  $K$ , а чисел в нем содержалось бы как можно меньше.

# Решение

Зафиксируем позицию первого из искомых чисел (левый указатель). Найдем минимальную позицию второго числа (правого указателя), где сумма чисел между ними будет больше  $K$  (основное условие). Запомним количество чисел в найденном отрезке – это наш текущий результат. Будем сдвигать левый указатель, пока основное условие не перестанет выполняться, на каждом шаге пытаемся улучшить результат. Снова фиксируем левый указатель и продолжаем выполнение алгоритма, пока сдвиги указателей возможны.

# Реализация

```
int res = 0;
int r = 0;
int s = 0;
int best_l = 0;
int best_r = r;
for (int l = 0; l < n; l++)
{
    if (s > k)
    {
        if (r - 1 < res)
        {
            res = r - 1;
            best_r = r;
            best_l = l;
        }
        s -= a[l];
        continue;
    }
    while (s <= k && r < n)
    {
        s += a[r];
        r++;
    }
}
```