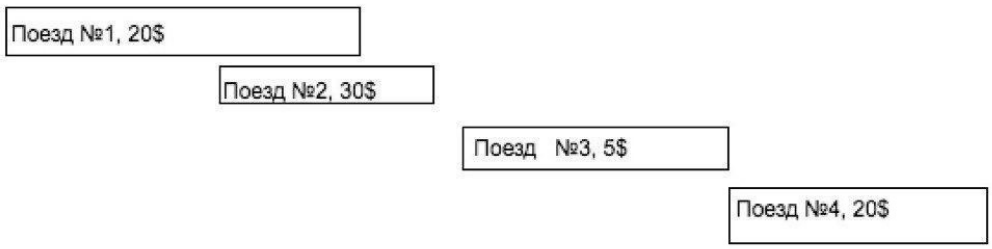
Переформулировка условия

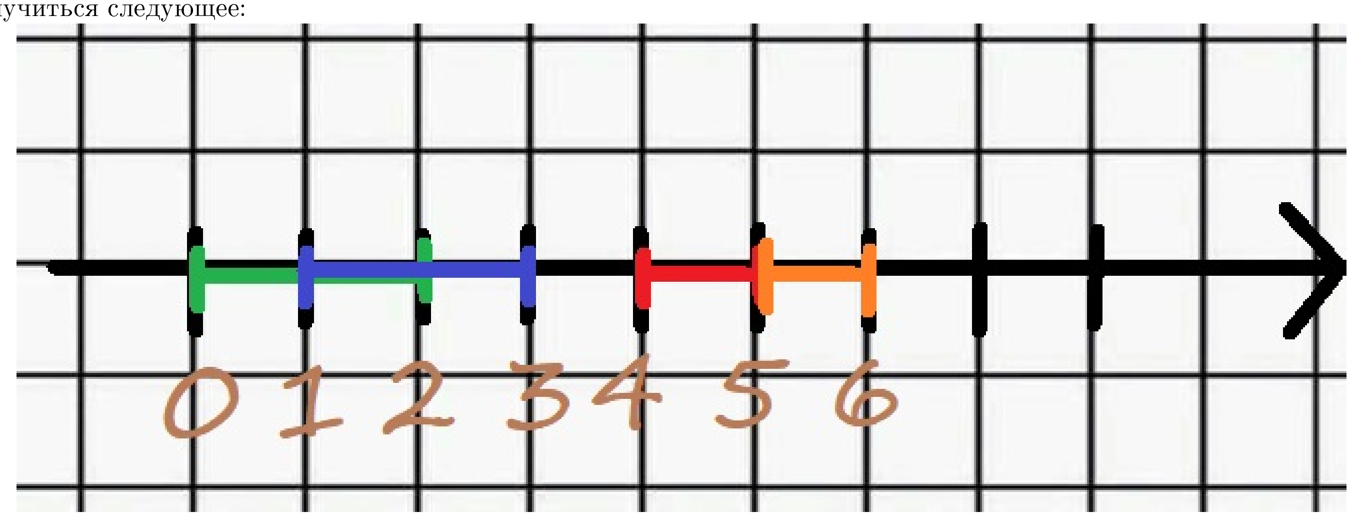
Переформулируем условие в удобную для восприятия форму. Будем считать, что вместо поездов у нас есть взвешенные отрезки (начало отрезка = время прибытия, конец отрезка = время прибытия + время на разгрузку), где веса - профит от разгрузки данного поезда. Тогда задача в том, чтоб найти подможество непересекающихся отрезков с максимальной возможной суммой весов (будем считать, что пересечение в 1 точке, как у оранжевого и красного отрезка, за пересечение не считается).



На примере с фотки видно, что максимальная сумма 30 + 5 + 20 = 55

Махинации с входными данными

Мы имеем дело с координатами концов отрезков, но непонятно каким типом данных задаются концы (может быть числом с плавающей точкой или в формате СС:ММ:ЧЧ и т.д. и т.п.), но нам это и не важно. Важно лишь то, что координаты концов можно между собой сравнивать. Поэтому, для упрощения, преобразуем координаты в целые неотрицательные числа так, чтоб все координаты концов отрезков стали целыми числами, но при этом знаки сравнения между каждой парой концов отрезков сохранились (ещё это называется сжатие коордиант). Для этого отсортируем множество концов отрезков по возрастанию, а потом поставим в соответствие i-той координате целое число i (имеется ввиду нуль индексация). Надо обратить внимание, что сортировать нужно лишь уникальные координаты, поэтому, например, координата конца красного отрезка и начала оранжевого из примера выше должна войти в массив лишь один раз. С примером выше должно случиться следующее:



Надо обратить внимание на то что максимальная возможная координата после такого сжатия будет превосходить количество отрезков не более чем в 2 раза, это будет важно в дальнейшем.

Алгоритм

Решать будем методом динамического программирования. Пусть dp_i - это ответ для данной задачи, если в рассмотрение входят только те отрезки, у которых координаты концов не превосходят i. Пусть так же X_i - множество отрезков, которые заканчиваются в точке i (в смысле самая правая точка им пранадлежащая имеет координату ровно i). Тогда $dp_0 = 0$, а для i > 0, dp_i считается следующим образом

$$dp_i = \max\left(dp_{i-1}, \max_{x \in X_i}(dp_{\text{начало отрезка } x} + w)\right)$$

в случае если X_i непустое множество и

$$dp_i = dp_{i-1}$$

в случае если пустое.

Тогда нужный нам ответ лежит в dp_k, где k-максимальнаа координата среди концов отрезков.

Асимптотика

Пусть количество отрезков будет n.

В части со сжатием координат нужно создать массив из не более чем 2n элементов, отсортировать, а потом с помощью хеш таблицы заменить координаты концов на целые. Если использовать сортировку за $O(n \log n)$, то и время работы алгоритма будет $O(n \log n)$ в среднем с $O(n)$ дополнительной памяти.

В части с подсчетом динамики нужно выполнить не более $2n + \sum_{i=0}^{2n} |X_i|$ операций, но $\sum_{i=0}^{2n} X_i \leq 2n$, поэтому тут асимптотика $O(n)$ гарантированно с $O(n)$ дополнительной памяти.

Тогда в общем алгоритм работает за $O(n \log n)$ с $O(n)$ дополнительной памяти. Асимптотически это около идеального, а скорее всего вообще лучшее что может быть, если говорить про чистую зависимость от n.