

Тестовое задание JetBrains

Антон Бричёв

2020

1 Постановка задачи

1.1 Условие задания

Представьте себя грузчиком работающим на товарной станции. В течении дня на станцию прибывают поезда и их надо немедленно разгружать.

Для каждого поезда известно:

- номер поезда,
- время прибытия,
- время, которое у вас займёт разгрузка,
- сумма, которую вам заплатят за разгрузку этого поезда.

Начав разгрузку поезда, вы должны ее закончить и не можете разгружать два поезда одновременно. Необязательно браться за разгрузку всех поездов. На станции есть другие грузчики. Ваша задача написать алгоритм, который по этим исходным данным максимизирует ваш заработок.

1.2 Формализация задачи

1.2.1 Входные данные

В первой строке входного файла записано целое число поездов N . Последующие N строк описывают поезда, по одному в строке. Про каждый поезд известно: номер, время прибытия в формате hh:mm (в интервале с 00:00 по 23:59), время на разгрузку в формате hh:mm (в интервале с 00:00 по 23:59), плата за разгрузку.

Номер поезда является целым числом.

Поезда не обязаны быть упорядоченными по времени прибытия.

1.2.2 Выходные данные

На консоль выводится максимальная прибыль, которая может быть получена при разгрузке.

1.2.3 Пример

input.in	System.out
4 1321 17:30 01:20 1000 1422 13:50 00:25 500 1 13:45 00:06 350 480 12:10 00:45 450	1950

Грузчик начнет разгружать 480 поезд в 12:10 и закончит в 12:55, отдохнет и пропустит поезд с номером 1, так как после него не успеет разгрузить более прибыльный 1422, после 1422 разгрузит 1321 и в сумме заработает 1950 у.е.

2 Решение

2.1 Описание решения

Поезда приходят друг за другом, а время имеет свойство течь вперед, следовательно отсортировав поезда по времени прибытия их можно представить в виде ориентированного ациклического графа.

- Ассоциируем поезд с вершиной графа.
- Ориентированное ребро (u, v) между двумя вершинами будет присутствовать если мы успеваем разгрузить поезд u к прибытию поезда v .
- Добавим две фиктивные вершины s (исток) и t (сток):
 - для каждой вершины v добавим ребро (s, v)
 - для каждой вершины v добавим ребро (v, t)

У нас появился ориентированный граф из поездов, где каждая вершина(поезд) имеет свой вес(прибыль от разгрузки). Задача сводится к поиску простого пути из s в t с максимальной суммой весов вершин.

Поскольку граф ациклический мы можем применять топологическую сортировку. Топологически отсортируем вершины и применим подход динамического программирования для поиска максимальной прибыли.

Инициализация динамики:

- $d[s] = 0$
- $d[t] = 0$
- $\forall v \in Vertices \ d[v] = v.value + \max(d[u]) : (v, u) \in Edges$

Запустимся из вершины t и в порядке топологической сортировки дойдем до вершины s в $d[s]$ будет находится ответ.

2.2 Сложность алгоритма

Для определения вычислительной сложности попробуем оценить число ребер в получившемся графе.

Для каждой вершины мы добавляем минимум два ребра в фиктивные вершины. Предположим, что поезда идут друг за другом и мы соединим первый поезд со всеми остальными $n - 1$, следующий с $n - 2$ и так далее.

Для каждой из n вершин получится $O(n)$ ребер.

Топологическая сортировка $O(n + m)$.

Динамика смотрит все ребра.

Общая сложность $O(n^2)$.

2.3 Ограничения (можно реализовать)

Добавить поддержку секунд.

Выводить номера поездов, которые нужно разгружать.