
Разбор задачи «Сериял»

Автор задачи:	Михаил Путилин
Подготовка условия, решения и тестов:	Михаил Путилин
Автор разбора:	Михаил Путилин

Отсортируем размеры файлов по неубыванию, пусть $s_i \leq s_{i+1}$. Размеры файлов — s_1, \dots, s_n , для удобства дальнейших рассуждений положим $s_0 = 0$.

Пусть запросы — x_1, \dots, x_t , обозначим как y_i их частичные суммы: $y_i = x_1 + \dots + x_i$ ($0 \leq i \leq t$). При этом $y_0 = 0$, так как мы должны скачать все файлы целиком, а скачивать больше, чем размер наибольшего файла, бессмысленно, то $y_t = s_n$.

Заметим, что имеет смысл делать запросы только так, чтобы после каждого запроса суммарное число загруженных байт было равно размеру какого-то файла. Иными словами, если запросы — x_1, \dots, x_t , то для любого i от 0 до t должно выполняться $y_i = s_w$ для некоторого w .

Доказательство: предположим обратное. Пусть $s_w < y_i < s_{w+1}$, при этом возьмём наименьшее такое i . Тогда можно присвоить y_i значение s_w , уменьшив соответствующим образом x_i и увеличив x_{i+1} . Ответ от этого не ухудшится, так как теперь при загрузке файлов размера s_w в последнем пакете не будет лишних байт, а для остальных файлов ничего не изменится. Если в результате окажется, что y_i совпало с y_{i-1} , то y_i нужно будет вообще убрать.

Будем теперь решать задачу методом динамического программирования. Пусть есть последовательность запросов суммарного размера s_i . Тогда dp_i — это наименьший возможный суммарный размер пакетов, которые будут загружены, если скачивать при помощи этой последовательности все серии. При этом файлы размера не больше s_i будут скачаны целиком, а от остальных — только первые s_i байт.

Начальное значение: $dp_0 = 0$, так как $s_0 = 0$. Ответом будет dp_n .

Как вычислить dp_i : переберём последний запрос x_t . Как следует из изложенного ранее, он должен быть таким, что $s_i - x_t = s_w$ для некоторого w . Поэтому будем перебирать w от 0 до $i - 1$. Первые $t - 1$ запросов скачают s_w байт, потратив на это dp_w . Последний запрос будет выполнен только для файлов размера больше s_w , и для каждого из них будет скачано $x_t + k$. Значит, всего будет скачано $dp_w + cnt_w \cdot (s_i - s_w + k)$. dp_i — это минимум этих значений по всем w .

Время работы решения — $O(n^2)$. Заметим, что его можно далее оптимизировать до времени $O(n \log n)$ с использованием Convex Hull Trick, но при заданных ограничениях этого не требуется.