

# Лучший бегун

Автор задачи: Сергей Князевский

Разработчик задачи: Алексей Дацковский

Сначала разберём подгруппы, в которых не используется основная идея из общего решения:

**Подгруппа 2:** Всегда выиграет первый бегун: он должен после каждой дорожки перемещаться на одну влево (пока это возможно). Так он пробежит больше всех дорожек, так как он бегал по максимально коротким дорожкам.

**Подгруппа 3:** Можно решить с помощью динамического программирования.  $dp[i][j]$  (где  $i$  — текущая дорожка бегуна, а  $j$  — оставшееся время) равно максимальному количеству дорожек, которое можно пробежать, начиная с такой ситуации.

**Подгруппа 5:** Всегда выигрывает бегун, который начинает на дорожке с минимальной длиной, так как он может пробежать максимальное число раз по самой короткой дорожке.

Для остальных подгрупп нужно понять, как будет выглядеть оптимальный маршрут бегуна. Пусть он начинает на дорожке  $i$ .

- Он должен закончить на самой короткой дорожке, на которой он был (обозначим её  $j$ ). Если бы он остался на самой короткой дорожке и бегал по ней до конца времени, то пробежал бы не меньше дорожек, чем в любом другом сценарии.
- Также он должен максимально быстро перейти от  $i$  до  $j$ , и потом бегать только по  $j$ . Так он пробежит максимальное число дорожек среди всех сценариев.

Пусть у нас есть бегун с начальной дорожкой  $i$  и конечной дорожкой  $j$ . Определим количество дорожек, которое он пробежит. Не умаляя общности, пусть  $i < j$ . Тогда он потратит  $a_i + a_{i+1} + \dots + a_{j-1}$  времени, чтобы дойти до дорожки  $j$ , а потом оставшееся время будет бегать по  $j$ . Значит, он пробежит  $j - i + \lfloor \frac{T - (a_i + a_{i+1} + \dots + a_{j-1})}{a_j} \rfloor$  дорожек (если время от дорожки  $i$  до дорожки  $j$  не превышает  $T$ ). Если поддерживать префиксные суммы массива  $a$ , то мы сможем вычислять количество дорожек за  $O(1)$ .

Этого достаточно, чтобы решить **подгруппу 1**: для каждой начальной позиции перебрать все конечные позиции и найти максимальное число дорожек.

**Подгруппа 4:** здесь нужно перебирать в качестве конечных позиций только те дорожки, которые строго короче всех дорожек между начальной и ней. Так как длины дорожек не превосходят 20, то для каждой начальной позиции мы рассмотрим не более 20 конечных дорожек слева и справа. А чтобы быстро находить ближайшую слева (или справа) дорожку короче текущей, можно предположить их заранее.

**Подгруппа 6:** осталось применить небольшой трюк. Будем наоборот для каждой конечной позиции  $j$  находить бегуна, который пробежит максимальное число дорожек и закончит на дорожке  $j$ . Можно показать, что это будет либо ближайший бегун слева от дорожки, либо ближайший бегун справа (или бегун, начинающий на этой дорожке, если такой есть) — другие бегуны потратят больше времени на то, чтобы добраться до  $j$  (если  $j$  — оптимальная конечная позиция). Значит, нам нужно перебрать не более 2 начальных дорожек для каждой конечной дорожки.