

Стать сильнее

Автор задачи и разработчик: Даниил Орешников

Каждое описанное в задаче устройство — стек, то есть чем раньше в него добавляется компонент, тем позже он будет вынут. Таким образом, компоненты, добавленные раньше, будут находиться в устройстве строго дольше, чем те, которые были добавлены позже.

Осталось только заметить, что если $|a_i - a_j| \leq 1$, компоненты i и j не могут находиться в одном устройстве. Если между любыми двумя действиями с устройством должно пройти не меньше секунды, то между моментами их добавления прошло не меньше секунды, и между моментами их вынимания прошло не меньше секунды, таким образом соседние в устройстве компоненты должны иметь время работы, отличающееся хотя бы на 2.

Для решения первых двух подзадач достаточно было написать полный перебор распределения компонентов по устройствам. Более того, в первой подзадаче можно было просто рассмотреть несколько возможных случаев, чтобы определить, хватит ли одного, двух или трех устройств.

Ограничения третьей подзадачи гарантируют, что a_i даны в возрастающем порядке. Таким образом, не бывает совпадающих элементов, но могут присутствовать элементы, отличающиеся на 1. Достаточно проверить, существует ли такое i , что $a_{i+1} = a_i + 1$. Если такое есть, понадобится хотя бы два устройства, и двух достаточно — можно распределить все компоненты с четным временем работы в порядке его убывания в первое устройство, а все с нечетным — во второе. Если же $a_{i+1} - a_i > 1$ для всех i , то достаточно одного устройства.

Для решения всех оставшихся подзадач в начале необходимо отсортировать все a_i по возрастанию или по убыванию. В подзадаче номер 4 гарантируется, что все a_i четны, а значит необходимо и достаточно, чтобы компоненты с совпадающим временем работы были в разных устройствах. В отсортированном массиве найти максимальное количество одинаковых элементов можно за линейное время.

Для оставшихся двух подгрупп требовалось заметить, что если какие-то k подряд элементов в отсортированном массиве a_i отличаются друг от друга не более, чем на 1, но при этом $a_{i+k} - a_i > 1$ для всех i , то ровно k устройств будет необходимо достаточно — поместим в первое устройство компоненты с индексами $1, k + 1, 2k + 1, \dots$, во второе — с индексами $2, k + 2, 2k + 2, \dots$, и так далее, в последнем устройстве будут компоненты с индексами $k, 2k, 3k, \dots$.

Можно было найти такое k за время $\mathcal{O}(n^2)$, просто двигаясь вперед от каждого i по отдельности. А можно было и за линейное время, заметив, что чем больше i , тем больше будет первое j , для которого $a_j - a_i > 1$. Это означает, что можно применить метод двух указателей — искать j , соответствующее левому индексу $i + 1$, начиная с j , найденного на предыдущем шаге для левого индекса i . Метод двух указателей работает за линейное время, итоговое время работы — $\mathcal{O}(n \log n)$ из-за сортировки.