

Рисохранилище

В сельской местности находится длинная прямая дорога, которая известна как Рисовый Путь. Вдоль этой дороги расположены R рисовых полей. Каждое поле имеет целочисленную координату от 1 до L включительно. Рисовые поля задаются в порядке неубывания их координат. Формально, обозначим для $0 \leq i < R$ координату рисового поля с номером i как $X[i]$. Гарантируется, что $1 \leq X[0] \leq \dots \leq X[R-1] \leq L$.

Отметим, что *несколько рисовых полей могут иметь одинаковую координату*.

Планируется построить одно *рисохранилище*, в которое требуется завезти с полей как можно больше риса. Как и рисовые поля, *рисохранилище должно иметь целочисленную координату* от 1 до L включительно. Рисохранилище разрешается строить в любой целочисленной координате, в том числе и в координате, в которой уже есть одно или более рисовых полей.

С каждого рисового поля в течение каждого сезона снимают урожай, который помещается *ровно в 1 грузовике*. Чтобы доставить урожай в рисохранилище, необходимо нанять водителя грузовика. Стоимость работы водителя по перевозке груза на единицу расстояния составляет 1 бат. Другими словами, стоимость транспортировки риса от заданного поля до рисохранилища равна модулю разности их координат.

К сожалению, бюджет рассматриваемого сезона ограничен: нельзя потратить более B бат на транспортировку. Необходимо построить рисохранилище в таком месте, чтобы можно было завезти в него как можно больше риса.

Задание

Написать процедуру `besthub(R,L,X,B)`, которой передаются следующие параметры:

- R — количество рисовых полей. Поля пронумерованы от 0 до $(R-1)$.
- L — максимальная координата.
- X — одномерный массив целых чисел, отсортированный в порядке неубывания. Для каждого i ($0 \leq i < R$) рисовое поле с номером i имеет координату $X[i]$.
- B — бюджет.

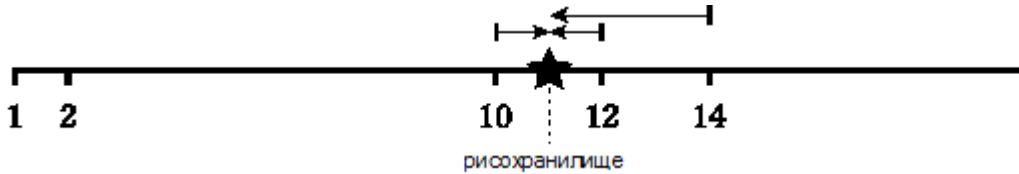
Ваша процедура должна находить оптимальное расположение рисохранилища и возвращать максимальное количество грузовиков риса, которые могут быть перевезены в рисохранилище, не превышая заданный бюджет.

Необходимо обратить внимание, что цена транспортировки риса может быть очень большой. Бюджет задаётся 64-битным целым числом, и поэтому рекомендуется использовать 64-битные целые числа для вычислений. В языках C/C++ используйте тип `long long`; в языке Паскаль используйте тип `Int64`.

Пример

Рассмотрим пример, в котором $R=5$, $L=20$, $B=6$, и

1
2
X= 10
12
14



В этом примере существует несколько оптимальных местоположений рисоохранилища. Можно поместить его в любой точке с целыми координатами от 10 до 14 включительно. На рисунке выше показано одно из этих оптимальных местоположений. В этом случае можно перевезти рис от полей в точках с координатами 10, 12 и 14 до рисоохранилища. Для каждого из этих оптимальных расположений общая стоимость транспортировки будет не более 6 бат. Очевидно, что никакое из возможных расположений рисоохранилища не позволит собирать рис более чем с трёх полей, таким образом, это решение оптимально и процедура besthub должна возвращать число 3.

Подзадачи

Подзадача 1 (17 баллов)

- $1 \leq R \leq 100$
- $1 \leq L \leq 100$
- $0 \leq B \leq 10\,000$
- Никакие два рисовых поля не имеют одинаковую координату (только для этой подзадачи!).

Подзадача 3 (26 баллов)

- $1 \leq R \leq 5\,000$
- $1 \leq L \leq 1\,000\,000$
- $0 \leq B \leq 2\,000\,000\,000$

Подзадача 2 (25 баллов)

- $1 \leq R \leq 500$
- $1 \leq L \leq 10\,000$
- $0 \leq B \leq 1\,000\,000$

Подзадача 4 (32 балла)

- $1 \leq R \leq 100\,000$
- $1 \leq L \leq 1\,000\,000\,000$
- $0 \leq B \leq 2\,000\,000\,000\,000\,000$

Детали реализации

Ограничения

- Ограничение по времени: 1 секунда
- Ограничение по памяти: 256 МВ

Замечание: Нет отдельного ограничения на размер стека; используемая стеком память входит в общий объём используемой памяти.

Интерфейс (API)

- Папка для разработки: `ricehub/`
- Участник должен разработать: `ricehub.c` или `ricehub.cpp` или `ricehub.pas`
- Интерфейс участника: `ricehub.h` или `ricehub.pas`
- Предлагаемый модуль оценивания: `grader.c` или `grader.cpp` или `grader.pas`
- Ввод для предлагаемого модуля оценивания: `grader.in.1`, `grader.in.2`, ...

Замечание: Предлагаемый модуль оценивания читает входной файл в следующем формате:

- Строка 1: **R**, **L** и **B**.
- Строки от 2 до **R**+1: информация о местоположении рисовых полей; то есть, строка с номером **i**+2 содержит значение координаты **X[i]** для $0 \leq i < R$.
- Строка **R**+2: ожидаемое решение.
- Ожидаемый вывод для предлагаемого модуля оценивания: `grader.expect.1`, `grader.expect.2`, ... В этой задаче каждый из перечисленных файлов должен содержать только текст “**Correct.**”