Ульянцев Владимир Игоревич,
Царев Федор Николаевич,
Цыпленков Алексей Евгеньевич

ЗАДАЧА «ПРОИЗВОДСТВО БЕНЗИНА»

Этой статьей мы продолжаем цикл публикаций олимпиадных задач для школьников по информатике. Решение таких задач и изучение разборов поможет Вам повысить уровень практических навыков программирования и подготовиться к олимпиадам по информатике.

В этой статье рассматривается задача «Производство бензина», которая предлагалась на первой Интернет-олимпиаде базового уровня сезона 2010–2011 г. Сайт этой олимпиады находится по адресу http://neerc.ifmo.ru/school/io/.

УСЛОВИЕ ЗАДАЧИ

Для оснащения нового цеха по производству бензина компания «Нанонефть» объявила конкурс. На него подали заявки n поставщиков соответствующих производственных линий. Для каждой заявки заданы три числа:

A_i – стоимость производственной линии;

B_i – затраты на производство одной тонны бензина на этой линии;

C_i – цена, по которой произведенную на этой линии тонну бензина готовы купить клиенты.

Точкой окупаемости называется то количество бензина, которое требуется произвести на линии, чтобы его суммарная цена была равна сумме стоимости линии и затрат на его производство.

Вам, как исполняющему обязанности менеджера «Нанонефти» предстоит сделать выбор оптимальной заявки. А именно, необходимо выбрать один вариант оснащения цеха, при котором точке окупаемости соответствует наименьшее количество бензина.

Формат входного файла

В первой строке входного файла содержится число n – количество заявок ($1 \le n \le 10^5$).

В следующих n строках заданы по три целых числа A_i, B_i, C_i ($1 \le A_i, B_i, C_i \le 10^9$, $B_i < C_i$).

Формат выходного файла

В выходной файл выведите номер заявки, при выборе которой точке окупаемости соответствует наименьшее количество бен-

зина. При существовании нескольких оптимальных заявок следует вывести номер наименьшей из них.

Примеры входных и выходных данных

petrol.in	petrol.out
2 2 1 3 1 2 3	1
3 1 2 4 3 1 4 2 2 4	1

РАЗБОР ЗАДАЧИ

Рассмотрим некоторую заявку, и пусть числа, заданные для нее в условии, равны A, B и C. Если t – ее точка окупаемости, то t является решением уравнения $A + B \cdot t = C \cdot t$. Перепишем данное уравнение в более удобной форме: $A = (C - B) \cdot t$. Так как по условию задачи для любой заявки $B < C$ и $A > 0$, то данное уравнение всегда будет иметь решение, причем это решение в силу положительности коэффициентов уравнения также будет положительным: $t = A/(C - B)$.

Теперь необходимо для каждой заявки решить уравнение такого вида и среди всех заявок выбрать ту, которой соответствует минимальное решение. Заметим, что для нахождения ответа не обязательно хранить все данные из входного файла – каждую заявку можно обработать и сравнить полученное значение с минимальным сразу после считывания.

Приведем реализацию предложенного решения на языке Паскаль (см. листинг 1).

В условии задачи все параметры запросов ограничены 10^9. Поэтому переменная

Листинг 1. **Реализация алгоритма**

```
const
  eps = 1e-9;

var
  a, b, c : longint;
  n, i, best : longint;
  low, cur : real;

begin
  assign(input, "petrol.in");
  assign(output, "petrol.out");
  reset(input);
  rewrite(output);

  read(n);
  best := 0;
  low := 1e10;

  for i := 1 to n do begin
    read(a, b, c);
    cur := a / (c - b);

    if (low - cur > eps) then begin
      low := cur;
      best := i;
    end;
  end;

  writeln(best);

  close(input);
  close(output);
end.
```

`low` инициализирована значением 10^{10}, так как оно заведомо больше любой возможной точки окупаемости.

Время работы этого алгоритма составляет $O(n)$.

Обратим внимание на типичную ошибку, допускаемую при решении данной задачи. При нахождении точки окупаемости ошибочно использовать деление нацело. Пример – тест из двух запросов: $A_1 = 1$, $B_1 = 4$, $C_1 = 4$, $A_2 = 1$, $B_2 = 4$, $C_2 = 5$. При делении нацело точка окупаемости для обоих запросов равна единице, и лучшим будет назван первый запрос, так как он идет раньше. Однако на самом деле точка окупаемости первого запроса – 4/3, что больше точки окупаемости второго запроса, которая в точности равна единице. Отдельно стоит обратить на это внимание при использовании *C/C++*, *Java* и прочих языков, в которых синтаксис деления целых чисел совпадает с синтаксисом деления нацело. В этом случае при делении значения необходимо привести к вещественному типу.

Также часто возникают ошибки при работе с вещественными числами из-за потерь точности. Все вещественные числа в памяти хранятся лишь с некоторой точностью, которая сильно теряется при делении и умножении. Про потерю точности читайте в учебниках по языкам программирования, а также в книге [1]. Поэтому сравнивать два вещественных числа на равенство нельзя, так как проверка на равенство сравнивает все разряды чисел. О приемах работы с вещественными числами читайте в книге [2] в решении задачи «2D».

Литература

1. *Кнут Д.* Искусство программирования, том 1. Основные алгоритмы. М.: Вильямс, 2007.
2. *Меньшиков Ф.* Олимпиадные задачи по программированию. М.: Питер, 2007.

Члены жюри Интернет-олимпиад по информатике базового уровня:

Ульянцев Владимир Игоревич, студент четвертого курса кафедры «Компьютерные технологии» (КТ) СПбГУ ИТМО, член жюри ВКОШП,

Царёв Федор Николаевич – аспирант кафедры КТ СПбГУ ИТМО, чемпион мира по программированию среди студентов 2008 года,

Цыпленков Алексей Евгеньевич, студент второго курса кафедры КТ СПбГУ ИТМО.