

Тренировки по алгоритмам 5.0 от Яндекса — Занятие 4 (Бинарный поиск)

26 апр 2024, 11:45:07
старт: 22 мар 2024, 20:30:00
финиш: 29 мар 2024, 18:00:00
длительность: 6д. 21ч.
начало: 22 мар 2024, 20:30:00
конец: 29 мар 2024, 18:00:00

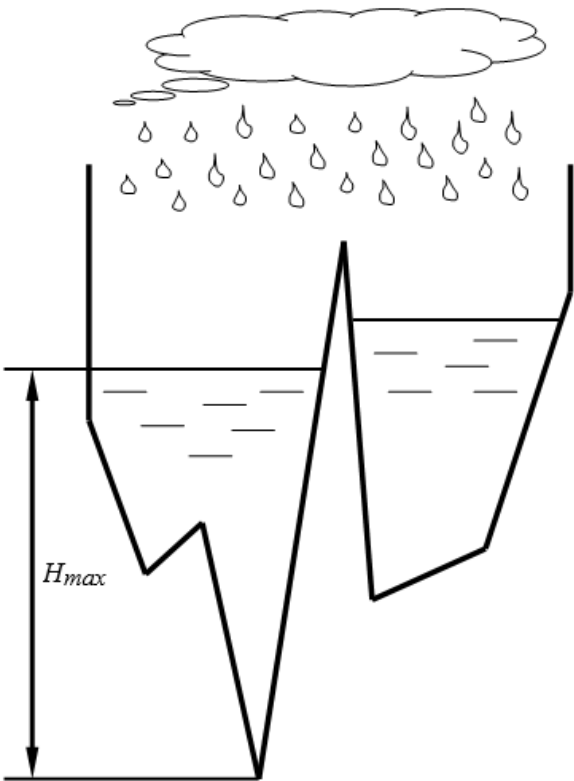
J. Дождик

Ограничение времени	2 секунды
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В НИИ метеорологии решили изучить процесс образования водоемов на различных рельефах местности во время дождя. Ввиду сложности реальной задачи была создана двумерная модель, в которой местность имеет только два измерения — высоту и длину. В этой модели рельеф местности можно представить как N -звенную ломаную с вершинами $(x_0, y_0), \dots, (x_N, y_N)$, где $x_0 < x_1 < \dots < x_N$ и $y_i \neq y_j$, для любых $i \neq j$. Слева в точке x_0 и справа в точке x_N рельеф ограничен вертикальными горами огромной высоты.

Если бы рельеф был горизонтальным, то после дождя вся местность покрылась бы слоем воды глубины H . Но поскольку рельеф — это ломаная, то вода стекает и скапливается в углублениях, образуя водоемы.

Требуется найти максимальную глубину в образовавшихся после дождя водоемах.



Формат ввода

В первой строке расположены натуральное число N ($1 \leq N \leq 100$) и H — действительное число, заданное с тремя цифрами после десятичной точки ($0 \leq H \leq 10^9$). В последующих $N + 1$ строках — по два целых числа x_i, y_i ($-10000 \leq x_i, y_i \leq 10000$). Числа в строках разделены пробелами.

Формат вывода

Выведите единственное число — искомую глубину с точностью 10^{-4} .

Пример

Ввод	Вывод
7 7.000 -5 10 -3 4 -1 6 1 -4 4 17 5 3 9 5 12 15	15.8446

Язык Python 3.9 (PyPy 7.3.11)

Набрать здесь Отправить файл

```
1 def waterright(now, ynow):
2     poly = [(x[now], ynow), (x[now], y[now])]
3     for j in range(now + 1, n + 2):
4         lastind = j
5
6         if y[j] > ynow:
7             break
8
9         poly.append((x[j], y[j]))
10
11     lastx = x[lastind - 1] + (x[lastind] - x[lastind - 1]) * (ynow - y[lastind - 1]) / (y[lastind] - y[lastind - 1])
12     poly.append((lastx, ynow))
13     poly.append((x[now], ynow))
14
15     square = 0 # площадь многоугольника
16     for i in range(len(poly) - 1):
17         square += poly[i][0] * poly[i + 1][1] - poly[i][1] * poly[i + 1][0]
18
19     square = abs(square)/2
20     water = (x[lastind] - x[now]) * h + sufadd[lastind]
21     return water - square
22
23 def waterleft(now, ynow):
24     poly = [(x[now], ynow), (x[now], y[now])]
25     for j in range(now - 1, -1, -1):
26         lastind = j
27
28         if y[j] > ynow:
29             break
30
31         poly.append((x[j], y[j]))
32
33     lastx = x[lastind + 1] + (x[lastind + 1] - x[lastind]) * (ynow - y[lastind + 1]) / (y[lastind] - y[lastind + 1])
34     poly.append((lastx, ynow))
35     poly.append((x[now], ynow))
36
37     square = 0
38
```

Отправить
Предыдущая