**Заколдованный аквариум. Решение.**

Можно рассмотреть все возможные уровни воды, начиная с 1. Так, если под уровнем воды нет ни одного отверстия, то вода будет подниматься дальше. Поэтому решение задачи целесообразно искать в промежутке начиная с уровня, где начинаются отверстия, до уровня, где они заканчиваются (miny, maxy). Для каждого из этого уровня нужно определить площадь отверстий, находящихся под водой b. Затем объем вытекающей жидкости W. Если этот объем меньше объема поступающей жидкости, то заполнение аквариума продолжается, и нужно рассматривать следующий уровень. Действия повторяются до тех пор, пока объем вытекающей жидкости не станет больше или равно объему поступающей.

Таким образом выполняется прямой поиск подходящего уровня воды.

V,M,N = map(int, input().split())

x=[0]\*N

y=[0]\*N

a=[0]\*N

maxy=0

for i in range(N):

x[i],y[i],a[i] = map(int, input().split())

if y[i]+a[i]>maxy:

maxy=y[i]+a[i] # наивысшая точка отверстия

miny=min(y)+1 #самое низкое отверстие

for k in range(miny,maxy+1):

f=0

b=0

for j in range(N):

if y[j]<k and y[j]+a[j]>=k:

b=b+(a[j]\*(k-y[j]))

elif y[j]<k and y[j]+a[j]<k:

b=b+a[j]\*a[j]

W=V\*b

if W>=M:

f=1

break

if f==1:

print(k)

else:

print(-1)

Для ускорения работы программы вместо прямого поиска можно использовать двоичный. Расстояние между минимальным и максимальным уровнем делится пополам. В этой срединной точке sr определяется объем вытекающей жидкости W. Если этот объем больше объема поступающей жидкости M, то искомый уровень находится ниже и maxy опускаем на уровень sr-1. Если же W меньше M, то аквариум будет заполняться, и искомый уровень будет находиться выше, min y поднимаем на уровень sr+1. Все действия продолжаются до тех пор пока не W=M, либо пока maxy не сравняется с miny.

Стоит отметить, что до начала поиска целесообразно проверить, не будет ли вода переливаться из аквариума.

import math

V,M,N = map(int, input().split())

x=[0]\*N

y=[0]\*N

a=[0]\*N

maxy=0

S=0

for i in range(N):

x[i],y[i],a[i] = map(int, input().split())

if y[i]+a[i]>maxy:

maxy=y[i]+a[i]

S=S+a[i]\*a[i]

miny=min(y)+1

if V\*S>M:

W=0

maxy=maxy-1

while miny!=maxy:

sr=math.floor((miny+maxy)/2)

b=0

for j in range(N):

if y[j]<sr and y[j]+a[j]>=sr:

b=b+(a[j]\*(sr-y[j]))

elif y[j]<sr and y[j]+a[j]<sr:

b=b+a[j]\*a[j]

W=V\*b

if W>M:

maxy=sr-1

elif W<M:

miny=sr+1

else:

ur=sr

break

ur=sr

elif V\*S==M:

ur=maxy

else:

ur=-1

print(ur)