АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ДЪРВОЯДИ

Означаваме $c_i=a_i/b_i$. Очевидно колкото е по-голямо това число, толкова по-дълго време ще се движи в i-тата плоскост първият дървояд в сравнение с втория. Следователно трябва да осигурим за първия дървояд тези плоскости, за които това число е по-голямо, а за втория – плоскостите, за които това число е по-малко. Пренареждаме плоскостите така, че стойностите на c_i да намаляват отляво-надясно и считаме, че първият дървояд навлиза отляво, а вторият – отдясно. След това в програмата извършваме моделиране на движението на двата дървояда, докато се срещнат. Очевидно може да считаме, че всички плоскости са с дебелина, равна на единица. Срещата може да се осъществи или на границата на две плоскости, или вътре в някоя плоскост. Във втория случай се пресмята времето за движение вътре в плоскостта, като се използва пропорция от скоростите на двата дървояда.

Доказателство за оптималността на решението:

Приемаме, че дебелината на всяка плоскост е 1. Означаваме с x_i частта от i-тата плоскост, в която първият дървояд се движи ($0 \le x_i \le 1$). Понеже скоростта на първия дървояд в тази плоскост е $1/a_i$, времето му на движение там е $a_i x_i$. Тогава сумарното време през всички плоскости за първия дървояд е:

$$\sum_{i=1}^{N} a_i x_i$$

За втория дървояд остава част, равна на $1-x_i$, която да измине в i-тата плоскост и понеже скоростта му там е $1/b_i$, времето му в тази плоскост е $b_i(1-x_i)$. Тогава сумарното време на втория дървояд е:

$$\sum_{i=1}^{N} b_i (1-x_i)$$

Понеже двата дървояда трябва да се движат еднакво време:

$$\sum_{i=1}^{N} a_i x_i = \sum_{i=1}^{N} b_i (1-x_i)$$

Задачата се свежда към задачата „Дробна раница”:

$$\max: \quad \sum_{i=1}^{N} a_i x_i$$

$$\sum_{i=1}^{N} (a_i + b_i) x_i = \sum_{i=1}^{N} b_i$$

$$0 \le x_i \le 1$$

Стандартния метод за решаване се състои в сортиране на коефициентите

$$\frac{a_i}{a_i + b_i}, \ i = 1, 2, ..., N$$

в намаляващ ред (което е еквивалентно със сортиране в намаляващ ред на a_i/b_i). След това в съответния на тази сортировка ред избираме индекс k, $1 \le k \le N$, и полагаме стойностите:

$x_1 = \ldots = x_{k-1} = 1$, $x_{k+1} = \ldots = x_N = 0$ и вземаме x_k ($0 \le x_k \le 1$) такова, че:

$a_1 + \ldots + a_{k-1} + a_k x_k = b_k(1-x_k) + b_{k+1} + \ldots + b_N$

Намирането (виж програмата) на k извършваме с линейна скорост по N, като започваме с `as`=0 и `bs`=сумата на всички b_i и на i-тата стъпка увеличаваме `as` с a_i и намаляваме `bs` с b_i. Стойността x_k намираме чрез решаване на линейно уравнение.

```
#include<iostream>
using namespace std;

const int N=1000;
int n;
struct ps {double a, b;};
ps p[N];

bool cmp(ps p1, ps p2)
{
 return p1.a/p1.b > p2.a/p2.b;
}

int main()
{
 cin >> n;
 for(int i=0;i<n;i++) cin >> p[i].a >> p[i].b;

 sort(p,p+n,cmp);

 double sa,sb,x,t;
 sa=0;
 sb=0;
 for(int i=0;i<n;i++) sb += p[i].b;

 int i=-1;;
 while(sa<=sb)
 {
  i++;
  sa += p[i].a;
  sb -= p[i].b;
 }
 if(sa==sb) t=sa;
 else
 {
  sa -= p[i].a;
  x=(sb-sa+p[i].b)/(p[i].a+p[i].b);
  t = sa+p[i].a*x;
 }
 cout << t << endl;
}
```

Автор: Зорница Дженкова