**Bartosz Tarnawski**
Treść zadania, Opracowanie

**Adam Karczmarz, Tomasz Syposz**
Program

**Dostępna pamięć: 128 MB.** *OI, etap III, dzień drugi, 3.04.2014*

# Panele słoneczne

*Bajtazar postanowił zainwestować w odnawialne źródła energii i założył fabrykę paneli słonecznych. Okazało się to trafnym posunięciem – już po kilku dniach do Bajtazara zgłosiło się n klientów. Każdy z nich zamówił jeden prostokątny panel, podając przy tym dopuszczalny zakres jego wysokości i szerokości.*

*Produkowane panele składają się z kwadratowych ogniw fotowoltaicznych. Dostępne są ogniwa dowolnych całkowitych rozmiarów, ale wszystkie ogniwa danego panelu muszą być jednakowe. Stosowany proces technologiczny powoduje, że im większe są ogniwa, z których składa się panel, tym jest on bardziej wydajny. Bajtazar chciałby zatem dla każdego z zamówionych paneli poznać maksymalną długość boku ogniw, z których można go wyprodukować.*

## Wejście

*Pierwszy wiersz standardowego wejścia zawiera jedną liczbę całkowitą* $n$ $(1 \leqslant n \leqslant 1000)$, *oznaczającą liczbę zamówionych paneli. W kolejnych* $n$ *wierszach znajdują się opisy poszczególnych paneli:* $i$*-ty z nich zawiera cztery liczby całkowite* $s_{min}, s_{max}, w_{min}, w_{max}$ $(1 \leqslant s_{min} \leqslant s_{max} \leqslant 10^9,\ 1 \leqslant w_{min} \leqslant w_{max} \leqslant 10^9)$ *pooddzielane pojedynczymi odstępami, oznaczające odpowiednio minimalną i maksymalną szerokość oraz minimalną i maksymalną wysokość* $i$*-tego panelu.*

*W testach wartych 75% punktów dla każdego panelu zachodzi dodatkowy warunek* $s_{max}, w_{max} \leqslant 10^7$. *W podzbiorze tych testów wartym 20% punktów zachodzi dodatkowy warunek* $n \leqslant 10$.

## Wyjście

*Twój program powinien wypisać na standardowe wyjście dokładnie* $n$ *wierszy, zawierających odpowiedzi do kolejnych przypadków testowych z wejścia: w* $i$*-tym wierszu ma znajdować się liczba całkowita oznaczająca maksymalną długość boku ogniw, z których można wyprodukować* $i$*-ty panel.*

## Przykład

*Dla danych wejściowych:*

```
4
3 9 8 8
1 10 11 15
4 7 22 23
2 5 19 24
```

*poprawnym wynikiem jest:*

```
8
7
2
5
```

**Wyjaśnienie do przykładu:** *Bajtazar wyprodukuje cztery panele słoneczne o następujących rozmiarach:* $8 \times 8$ *(złożony z jednego ogniwa),* $7 \times 14$ *(złożony z dwóch ogniw),* $4 \times 22$ *lub* $6 \times 22$ *(złożony z 22 lub 33 ogniw) oraz* $5 \times 20$ *(złożony z czterech ogniw).*

**Testy „ocen”:**

`1ocen`: *$n = 1000$, $s_{max}, w_{max} \leqslant 10^7$; dla każdego z zamówionych paneli, maksymalna długość boku ogniw, z których można go wyprodukować, jest równa $s_{max}$.*

# Rozwiązanie

Powiemy, że liczba całkowita $k$ jest *zgodna* z parą przedziałów o całkowitych końcach $([a,b],\ [c,d])$, jeśli istnieją dla niej całkowite $x, y$ takie, że $kx \in [a,b]$ oraz $ky \in [c,d]$.

Dane jest $n$ par przedziałów o współrzędnych końców pomiędzy 1 a $m$. Dla każdej z nich chcemy znaleźć największą liczbę zgodną z tą parą przedziałów.

## Rozwiązanie wzorcowe $O(n\sqrt{m})$

Nasze rozwiązanie będzie działało *on-line* (tzn. na zapytania będziemy odpowiadać niezależnie) i wystarczy mu stała ilość pamięci.

Ustalmy parę przedziałów $([a,b],[c,d])$, nazwijmy ją $P$. Będziemy potrzebowali kilku obserwacji.

**Lemat 1.** Liczba $k$ jest zgodna z $P$ wtedy i tylko wtedy, gdy $\left\lceil \frac{a}{k} \right\rceil \leqslant \left\lfloor \frac{b}{k} \right\rfloor$ oraz $\left\lceil \frac{c}{k} \right\rceil \leqslant \left\lfloor \frac{d}{k} \right\rfloor$.

**Dowód:** Wystarczy zauważyć, że dla liczb całkowitych dodatnich $k, x, b$ mamy równoważność: $kx \leqslant b \iff x \leqslant \left\lfloor \frac{b}{k} \right\rfloor$. Podobnie $kx \geqslant a \iff x \geqslant \left\lceil \frac{a}{k} \right\rceil$. ■

**Lemat 2.** Załóżmy, że $k$ jest największą liczbą zgodną z $P$, $kx \in [a,b]$ oraz $ky \in [c,d]$. Wtedy $k = \left\lfloor \frac{b}{x} \right\rfloor$ lub $k = \left\lfloor \frac{d}{y} \right\rfloor$.

**Dowód:** Wiemy, że $k+1$ nie jest dobra, stąd $(k+1)x > b$ lub $(k+1)y > d$. Bez straty ogólności możemy założyć, że $(k+1)x > b$, a wtedy nierówności $k \leqslant \frac{b}{x}$ i $(k+1) > \frac{b}{x}$ implikują $k = \left\lfloor \frac{b}{x} \right\rfloor$. ■

Trzecia obserwacja wydaje się całkiem prosta, a jednak jest kluczem do rozwiązania zadania.

**Lemat 3.** Niech $k, x, y$ będą jak w założeniach lematu 2 oraz niech $m = \max(b,d)$. Wówczas (1) $k \leqslant \sqrt{m}$ lub (2) $x, y \leqslant \sqrt{m}$.

**Dowód:** W przeciwnym razie $kx > b$ lub $ky > d$. ■

Teraz możemy już opisać wzorcowy algorytm:

**function** najlepsze_ogniwo($P = ([a,b],[c,d])$)
**begin**
  $m := \max(b,d)$;
  $wynik := 1$;
  { Przypadek 1: $wynik \leqslant \sqrt{m}$ }

**for** $k := 2$ **to** $\lfloor\sqrt{m}\rfloor$ **do**
  **if** $k$ zgodna z $P$ **then** { użyj lematu 1 }
    $wynik := k$;

{ Przypadek 2: $wynik \geqslant \sqrt{m}$ }
**for** $l := 1$ **to** $\lfloor\sqrt{m}\rfloor$ **do begin**
  $k_1 := \left\lfloor \frac{b}{l} \right\rfloor$;
  $k_2 := \left\lfloor \frac{d}{l} \right\rfloor$;
  **for** $i$ **in** $\{1, 2\}$ **do**
    **if** $k_i$ zgodna z $P$ **then** { użyj lematu 1 }
      $wynik := \max(wynik, k_i)$;
**end**
**return** $wynik$;
**end**

Został on zaimplementowany w plikach `pan.cpp`, `pan1.pas`.

## Rozwiązanie alternatywne $O(n\sqrt{m})$

Przypadek 2 możemy obsługiwać także w trochę inny sposób. Nie musimy korzystać z charakteryzacji największej zgodnej liczby z lematu 2. W zamian przychodzi nam z pomocą:

**Lemat 4.** Niech $x, y$ będą liczbami całkowitymi. Oznaczmy $I_x = \left[\left\lceil \frac{a}{x} \right\rceil, \left\lfloor \frac{b}{x} \right\rfloor\right]$, $J_y = \left[\left\lceil \frac{c}{y} \right\rceil, \left\lfloor \frac{d}{y} \right\rfloor\right]$. Wówczas zbiór liczb całkowitych $k$ takich, że $kx \in [a, b]$ oraz $ky \in [c, d]$, można zapisać jako przecięcie $I_x \cap J_y$, ograniczone do liczb całkowitych.

Będziemy szukać największej liczby całkowitej, która należy do przecięcia $I_x \cap J_y$ dla pewnych $x, y \leqslant \sqrt{m}$. Poszukiwanie możemy zrealizować w czasie $O(\sqrt{m})$.

$x := 1$; $y := 1$;
**while** $x \leqslant \sqrt{m}$ **and** $y \leqslant \sqrt{m}$ **do begin**
  **if** $I_x \cap J_y \neq \emptyset$ **then**
    **return** $\max I_x \cap J_y$;
  **else if** $I_x$ leży na lewo od $J_y$ **then begin**
    $y := y + 1$;
    **while** $y \leqslant \sqrt{m}$ **and** $J_y = \emptyset$ **do**
      $y := y + 1$;
  **end else begin**
    $x := x + 1$;
    **while** $x \leqslant \sqrt{m}$ **and** $I_x = \emptyset$ **do**
      $x := x + 1$;
  **end**
**end**
**return** wynik mniejszy od $\sqrt{m}$, czyli zachodzi przypadek 1;

To rozwiązanie, tak samo jak wzorcowe, zużywa $O(1)$ pamięci. Zostało zaimplementowane w pliku `pan4.cpp`.

## Rozwiązanie wolniejsze $O(n\sqrt{m}\log m)$

Również to rozwiązanie różni się od wzorcowego tylko sposobem obsługi przypadku 2. W pewnym sensie robi to samo co rozwiązanie alternatywne, ale korzystając z abstrakcji „zamiatania".

Naszymi zdarzeniami będą lewe i prawe końce niepustych przedziałów typu $I_x$ oraz $J_y$ (zdefiniowanych w lemacie 4) dla $x, y \leqslant \sqrt{m}$. Mamy zatem cztery rodzaje zdarzeń (2 typy przedziałów $\times$ 2 końce odcinka).

Wrzucamy punkty do tablicy w dowolnej kolejności, a potem ją sortujemy; to zajmie nam $O(\sqrt{m}\log\sqrt{m}) = O(\sqrt{m}\log m)$ czasu. Następnie przeglądamy końce w kolejności malejących współrzędnych aż do wykrycia pierwszego przecięcia przedziałów dwóch różnych typów.

To rozwiązanie działa w czasie $O(n\sqrt{m}\log m)$ i na zawodach zdobywało około 70 punktów. Przykładowa implementacja znajduje się w pliku `pans11.cpp`. W rozwiązaniu tym można by osiągnąć czas $O(n\sqrt{m})$, gdyby punkty każdego rodzaju generować od razu w dobrej kolejności, a następnie w celu uzyskania posortowanej listy zdarzeń wykonać trzy scalenia.

## Testy

Przygotowano 17 testów. Nie były grupowane, za to w obrębie pojedynczego testu występowały zapytania różnych rodzajów. Wykorzystano następujące typy zapytań:

- całkowicie losowe
- wynik jest duży
- przedziały są krótkie
- przedziały są krótkie, wynik jest mały
- wynik jest blisko początku jednego z przedziałów
- wynik na pewno nie należy do żadnego z przedziałów
- jeden przedział jest krótki, drugi długi
- $([cp, cp], [a, b])$, gdzie $p$ jest liczbą pierwszą, $p > a, b$
- $([cp, cp], [a, b])$, gdzie $c \leqslant 8$, $p$ jest liczbą pierwszą, $cp < a, b$ lub $cp > a, b$
- $([a, b], [c, d])$, gdzie $a \leqslant c \leqslant b \leqslant d$ (przedziały mają wspólny punkt)
- $([a, b + t], [c, d + t])$, gdzie $b$ jest wynikiem, $b$ nie dzieli żadnej z liczb $d + 1, d + 2, \ldots, d + t$.