Zbigniew Czech
Treść zadania

Marcin Pilipczuk
Opracowanie, Program

OI, Etap III, dzień 2, 7–04–2005

Lustrzana pułapka

Lustrzana pułapka to prostopadłościan zbudowany z luster, których powierzchnie odbijające są skierowane do wnętrza prostopadłościanu. Dokładnie w środku prostopadłościanu zawieszony jest miniaturowy laser o rozmiarach punktu. Zadanie polega na takim skierowaniu lasera, aby jego promień przebył jak najdłuższą drogę i trafił w laser. Przy czym przez długość drogi rozumiemy sumę odległości, jaką przebył promień lasera w każdym z trzech kierunków równoległych do krawędzi luster (czyli mierzymy tzw. metryką miejską).

Wymiary lustrzanej pułapki są parzystymi liczbami całkowitymi. Krawędzie oraz wierzchołki, w których lustra stykają się ze sobą, nie odbijają promieni lasera. Wewnątrz pułapki określamy układ współrzędnych: osie układu są równoległe do krawędzi pułapki, a laser znajduje się w początku układu współrzędnych. Laser można skierować na dowolny punkt wewnątrz pułapki o współrzędnych całkowitych, włączając w to punkty na powierzchni luster (z wyjątkiem samego lasera, czyli punktu $(0,0,0)$).

Zadanie

Napisz program, który:

- *wczyta ze standardowego wejścia wymiary lustrzanej pułapki,*
- *wyznaczy kierunek takiego ustawienia lasera, że promień wystrzelony przez laser:*
  - *będzie odbijał się od luster, choć niekoniecznie od wszystkich,*
  - *nie trafi w krawędź ani w wierzchołek pułapki,*
  - *trafi ponownie do lasera, choć być może z innego kierunku,*
  - *przebędzie możliwie najdłuższą drogę (w sensie wyżej zdefiniowanej odległości).*
- *wypisze wynik na standardowe wyjście.*

Wejście

Pojedynczy test składa się z wielu lustrzanych pułapek do przeanalizowania. W pierwszym wierszu wejścia podana jest jedna liczba całkowita $1 \leqslant K \leqslant 1000$, oznaczająca liczbę pułapek do rozpatrzenia. W wierszach $2 \ldots K+1$ opisane są pułapki, po jednej w wierszu. Opis pułapki składa się z trzech liczb $5 \leqslant x,y,z \leqslant 1000$, oddzielonych pojedynczymi odstępami. Lustrzana pułapka ma wymiary $2x \times 2y \times 2z$.

Wyjście

Twój program powinien wypisać dokładnie K wierszy. W wierszu o numerze i powinno się znaleźć rozwiązanie dla pułapki o numerze i: trzy liczby całkowite k_x, k_y, k_z, oddzielone *pojedynczymi odstępami,* $-x \leqslant k_x \leqslant x$, $-y \leqslant k_y \leqslant y$, $-z \leqslant k_z \leqslant z$, $(k_x, k_y, k_z) \neq (0,0,0)$. *Liczby te oznaczają, że w i-tej pułapce laser powinien być skierowany na punkt o współrzędnych* (k_x, k_y, k_z).

Jeżeli istnieje wiele poprawnych wyników, Twój program powinien wypisać dowolny z nich.

Przykład

Dla danych wejściowych:

```
2
5 6 7
5 6 6
```

poprawnym wynikiem jest:

```
4 5 6
4 6 5
```

Rozwiązanie

Wstęp

Streszczenie

Rozwiązanie wzorcowe składa się z dwóch części.

Najpierw zajmiemy się sprawdzeniem, czy promień lasera skierowany w określony punkt powróci do środka pułapki nie trafiając wcześniej w żadną krawędź. Pokażemy najpierw algorytm działający w czasie $O(xyz)$ dla pułapki o rozmiarach $2x \times 2y \times 2z$. Następnie ulepszymy rozwiązanie przyśpieszając algorytm do czasu $O(\log(xyz))$. Algorytm dodatkowo będzie zwracał długość drogi, którą przebył promień.

Mając powyższy algorytm możemy spróbować przetestować wszystkie ustawienia lasera i sprawdzić, które z nich gwarantuje najdłuższą podróż promienia przed powrotem do punktu wyjścia. Takie rozwiązanie, działające w czasie $O(xyz \cdot \log(xyz))$, jest jednak zbyt wolne przy ograniczeniach podanych w zadaniu. Dlatego też w drugiej części rozwiązania zastanowimy się nad wyeliminowaniem niektórych ustawień. Pokażemy, że przeglądając je w odpowiedniej kolejności już po krótkim czasie będziemy mogli zakończyć poszukiwania, uznawszy, że lepszego rozwiązania nie znajdziemy.

Oznaczenia

Wprowadźmy następujące pomocnicze oznaczenia:

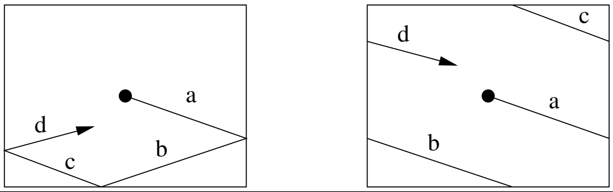
- $ord(n) = max\{k : 2^k \mid n\}$, czyli jest to maksymalna potęga dwójki, przez którą można podzielić liczbę n;
- punkt o nieujemnych współrzędnych całkowitych (a,b,c) nazwiemy *punktem pierwotnym* wtedy i tylko wtedy, gdy $NWD(a,b,c) = 1$.

Uproszczenia

Zauważmy, że zamiast rozważać promień lasera r odbijający się od ściany pułapki, możemy rozważać „promień zawinięty” $w(r)$ o stałym kierunku, który w momencie uderzenia w ścianę wychodzi dalej z odpowiedniego punktu na równoległej ścianie (patrz przykład dla przypadku dwuwymiarowego na rysunku 1). Choć promień $w(r)$ biegnie inną drogą niż oryginalny, to pod pewnymi względami jest mu równoważny:

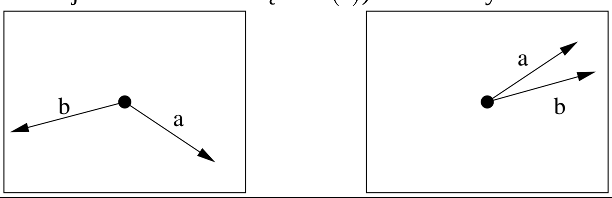
- promień r jest w chwili t w punkcie o współrzędnych całkowitych wtedy i tylko wtedy, gdy promień $w(r)$ jest w chwili t w punkcie o współrzędnych całkowitych;
- promień r trafia w krawędź w chwili t wtedy i tylko wtedy, gdy promień $w(r)$ trafia w krawędź w chwili t;
- promień r wraca do środka pułapki w chwili t po przebyciu drogi d wtedy i tylko wtedy, gdy promień $w(r)$ wraca do środka pułapki w chwili t po przebyciu drogi d.

Prawdziwość tych własności możemy łatwo uzasadnić, gdy tylko zauważymy, że punkty $P(t)$ i $P'(t)$ w których znajdują się odpowiednio promień r i promień $w(r)$ w chwili t, mają odpowiednie współrzędne równe z dokładnością do znaku. To sprawia, że możemy rozważać promień $w(r)$ zamiast promienia r.



Rys. 1: Oryginalny promień r oraz „promień zawinięty” $w(r)$. Dla czytelności odpowiadające sobie fragmenty drogi promienia r i $w(r)$ oznaczono takimi samymi etykietami.

Następnie zauważmy, że możemy ograniczyć się do rozważania promieni wystrzelonych z lasera skierowanego w punkty o współrzędnych nieujemnych (patrz przykład dla przypadku dwuwymiarowego na rysunku 2). Dla każdego innego promienia istnieje równoważny mu (w takim samym sensie jak równoważne są r i $w(r)$) skierowany właśnie w takim kierunku.



Rys. 2: Dwa dowolne promienie oraz równoważne im promienie skierowane na punkty o współrzędnych nieujemnych.

Wobec powyższego przyjmijmy następujące założenia:

- lewy-dolny róg pułapki znajduje się w punkcie $(-x,-y,-z)$, prawy-górny róg jest w punkcie (x,y,z), a laser jest umieszczony w punkcie $(0,0,0)$;
- punkty, których kolejne współrzędne różnią się o całkowitą wielokrotność liczb $2x$, $2y$ i $2z$, będziemy utożsamiać (tzn. punkt (a,b,c), dla $a \in (2ix-x, 2ix+x]$,

$b \in (2jy - y, 2jy + y]$ oraz $c \in (2kz - z, 2kz + z]$ dla pewnych całkowitych i, j oraz k, oznacza punkt o współrzędnych $(a - 2ix, b - 2jy, c - 2kz)$), a operacje arytmetyczne na kolejnych współrzędnych będziemy przeprowadzać cyklicznie odpowiednio w przedziałach: $(-x, x]$, $(-y, y]$ oraz $(-z, z]$;

- powiemy, że laser *strzela* w (a, b, c) jeśli uruchamiamy go po skierowaniu na ten właśnie punkt;
- rozważając laser strzelający w punkt (a, b, c) będziemy mówić, że jego promień pokonuje odległość od środka pułapki do punktu (a, b, c) (czyli według warunków zadania odległość $a + b + c$) w jednym *kroku*;
- powyższy promień będzie po t krokach (dla dowolnego t, także niecałkowitego i/lub niedodatniego) w punkcie o współrzędnych (ta, tb, tc);

Na drodze do algorytmu wzorcowego

Na początku zanotujmy kilka spostrzeżeń. Z przyczyn, które staną się jasne już za chwilę, spostrzeżenia te będą dotyczyły zarówno sytuacji w przestrzeni trójwymiarowej, jak i na płaszczyźnie — w przestrzeni dwuwymiarowej.

Lemat 1 *Rozważmy strzał w pułapce trójwymiarowej w punkt pierwotny (a, b, c). Punkt, w którym jest promień po t krokach, ma współrzędne całkowite wtedy i tylko wtedy, gdy t jest liczbą całkowitą.*

Dla pułapki dwuwymiarowej zachodzi analogiczna własność.

Dowód Po czasie t rozważany promień jest w punkcie o współrzędnych (ta, tb, tc). Oczywiście jeśli liczba t jest całkowita, to wszystkie współrzędne tego punktu też są całkowite. Załóżmy teraz, że liczby ta, tb i tc są całkowite. To oznacza, że liczba t musi być wymierna, czyli można ja przedstawić w postaci nieskracalnego ułamka $t = \frac{p}{q}$ i $NWD(p, q) = 1$. Ale stąd wynika, że liczba q dzieli liczbę a, a także liczby b i c. Widzimy więc, że $q = 1$, gdyż punkt (a, b, c) jest pierwotny i $NWD(a, b, c) = 1$.

Dowód można praktycznie bez zmian zastosować do przypadku dwuwymiarowego. ■

Wiosek 1

1. Jeśli po strzale w punkt (a, b, c) promień wróci do środka pułapki po przebyciu drogi d, to po strzale w punkt pierwotny $(\frac{a}{NWD(a,b,c)}, \frac{b}{NWD(a,b,c)}, \frac{c}{NWD(a,b,c)})$ promień wróci do środka pułapki po przebyciu tej samej drogi d.
2. Laser skierowany w punkt pierwotny (a, b, c) może powrócić do środka pułapki tylko po całkowitej liczbie kroków, czyli po przebyciu drogi $t(a + b + c)$ dla pewnej liczby całkowitej t.
3. Jeżeli po strzale w punkt (a, b, c) promień nie trafi po drodze w krawędź, to zawsze powróci do środka pułapki po wykonaniu nie więcej niż $2x \cdot 2y \cdot 2z$ kroków — tyle jest wszystkich punktów w pułapce.

Teraz pozostaje już tylko zbadać, czy po strzale w punkt pierwotny (a,b,c) promień nie ugrzęźnie na krawędzi. W tym celu dla promienia r oznaczmy przez $P_Z(r)$ jego rzut na ścianę pułapki prostopadłą do osi OZ. Zauważmy, że promień r trafia w jedną z krawędzi pułapki równoległych do osi OZ wtedy i tylko wtedy, gdy jego rzut $P_Z(r)$ trafia w jeden z wierzchołków ściany, na którą został zrzutowany. Stąd mamy kolejne spostrzeżenie, które pozwoli nam odszukać moment uderzenia promienia w krawędź, o ile ono nastąpi.

Wiosek 2

1. Promień r po strzale w punkt pierwotny (a,b,c) trafi w krawędź równoległą do osi OZ wtedy i tylko wtedy, gdy promień $P_Z(r)$ po strzale na płaszczyźnie XOY w punkt (a,b) trafi w punkt (x,y).

2. Promień $P_Z(r)$ biegnie taką samą drogą jak promień r' wystrzelony na płaszczyźnie XOY w punkt pierwotny $\left(\frac{a}{NWD(a,b)}, \frac{b}{NWD(a,b)}\right)$.

3. Promień r' może trafić w wierzchołek ściany tylko po całkowitej liczbie kroków.

Dokonane dotychczas spostrzeżenia pozwalają nam sformułować pierwszy algorytm rozwiązania zadania.

```
procedure Luss1;
  begin
    Tmax=0;
    for a=0 to x do
      for b=0 to y do
        for c=0 to z do
          if NWD(a,b,c)=1 then
            t=(a+b+c)·sprawdźStrzał(a,b,c);
            if t > Tmax then
              Tmax=t;
              (aOpt,bOpt,cOpt)=(a,b,c);
    if Tmax>0 then return (aOpt,bOpt,cOpt)
    else return NULL;
  end
```

```
procedure sprawdźStrzał(a,b,c)
  begin
    a1=a/NWD(a,b); b1=b/NWD(a,b);
    for t=1 to 2x·2y do
      if (ta1,tb1) = (x,y) then return 0;
    a1=a/NWD(a,c); c1=c/NWD(a,c);
    for t=1 to 2x·2z do
      if (ta1,tc1) = (x,z) then return 0;
    b1=b/NWD(b,c); c1=c/NWD(b,c);
    for t=1 to 2y·2z do
      if (tb1,tc1) = (y,z) then return 0;
    for t=1 to 2x·2y·2z do
      if (ta,tb,tc) = (0,0,0) then return t;
  end
```

Algorytm ten dla każdego punktu pierwotnego wykonuje test wymagający czasu xyz. Nie dokonując głębszej analizy liczności zbioru punktów pierwotnych dostajemy stąd ograniczenie złożoności czasowej algorytmu $O((xyz)^2)$. Teoretycznie nie jest więc to szybki algorytm — w praktyce także mieści się w limicie czasowym tylko na jednym z przygotowanych testów.

Przyśpieszenie

Zamiast śledzić bieg promienia i sprawdzać krok po kroku, czy nie powrócił on do środka pułapki, postaramy się wyliczyć moment trafienia promienia we wskazany punkt. Rozważmy strzał w punkt (a,b,c) — oczywiście pierwotny — i niech T oznacza liczbę kroków, po których tak wystrzelony promień po raz pierwszy powróci do środka pułapki, o ile nie zatrzyma go wcześniej żadna krawędź. Z lematu 1 wiemy, że T jest liczbą całkowitą. Podamy wzór, który pozwoli ją wyliczyć.

Lemat 2
$$T = NWW\left(\frac{NWW(2x,a)}{a}, \frac{NWW(2y,b)}{b}, \frac{NWW(2z,c)}{c}\right)$$
(jeśli któraś z liczb a, b lub c jest zerem, to przyjmiemy, że odpowiedni z argumentów NWW jest równy 1*).*

Dowód Liczbę T zdefiniowaliśmy jako liczbę kroków, po których promień wraca do środka pułapki, więc
$$(Ta,Tb,Tc) = (0,0,0),$$
czyli
$$Ta \equiv 0 \ \ mod\ \ 2x,\ \ Tb \equiv 0 \ \ mod\ \ 2y,\ \ Tc \equiv 0 \ \ mod\ \ 2z,$$
skąd
$$2x \mid Ta,\ \ 2y \mid Tb,\ \ 2z \mid Tc.$$
Mamy więc
$$NWW(2x,a) \mid Ta,\ \ NWW(2y,b) \mid Tb,\ \ NWW(2z,c) \mid Tc$$

i dalej

$$\frac{NWW(2x,a)}{a} \mid T, \quad \frac{NWW(2y,b)}{b} \mid T, \quad \frac{NWW(2z,c)}{c} \mid T,$$

czyli

$$NWW\Big(\frac{NWW(2x,a)}{a}, \quad \frac{NWW(2y,b)}{b}, \quad \frac{NWW(2z,c)}{c}\Big) \mid T.$$

Teraz wystarczy sprawdzić, że po T' krokach, gdzie

$$T' = NWW\Big(\frac{NWW(2x,a)}{a}, \frac{NWW(2y,b)}{b}, \frac{NWW(2z,c)}{c}\Big)$$

promień znajdzie się w środku pułapki (zakładając, że wcześniej nie trafi w żadną krawędź). Zauważmy, że

$$\frac{NWW(2x,a)}{a} \mid T', \text{ więc } NWW(2x,a) \mid aT', \text{ stąd } 2x \mid aT'.$$

Po analogicznym sprawdzeniu pozostałych współrzędnych widzimy, że

$$(T'a, T'b, T'c) \; = \; (0,0,0).$$

■

Poszukiwanie drogi przebytej przez promień z wykorzystaniem wyniku powyższego lematu przebiega w czasie $O(\log(xyz))$ — wystarczy do wyliczenia wartości T zastosować algorytm Euklidesa. Pamiętajmy jednak, że jest to tylko droga hipotetyczna, nie wiemy bowiem, czy promień nie trafi w krawędź przed powrotem do środka pułapki. Spróbujmy więc teraz podać wzór pozwalający rozstrzygnąć, czy promień trafi w krawędź.

Lemat 3 *Promień wystrzelony w punkt (a,b,c) trafi w krawędź równoległą do osi OZ wtedy i tylko wtedy, gdy $ord(a) - ord(b) = ord(x) - ord(y)$. Analogiczne implikacje zachodzą dla pozostałych krawędzi biegnących w pozostałych kierunkach.*

Dowód Rozważmy krawędź równoległą do osi OZ oraz rzut promienia lasera na współrzędne X i Y. Promień trafi w krawędź wtedy i tylko wtedy, gdy

$$\text{istnieje liczba rzeczywista } t \in (0,T): \; (at, bt) = (x,y) \tag{1}$$

Zauważmy, że powyższy warunek jest równoważny następującemu:

$$\text{istnieje liczba rzeczywista } t: \; (at, bt) = (x,y) \tag{2}$$

Oczywiście wystarczy uzasadnić implikację tylko w jedną stronę. Powiedzmy, że zachodzi warunek (2) i t jest liczbą, która go spełnia. Przypomnijmy, że promień wystrzelony w punkt (a,b,c), gdyby nie krawędzie, dotarłby po T krokach do środka pułapki, czyli

$$(Ta, Tb, Tc) = (0,0,0).$$

Stąd widzimy, że dla dowolnej liczby całkowitej i:

$$(at - i \cdot aT, bt - i \cdot bT) \; = \; (x,y).$$

Przyjmując $i = \lfloor t/T \rfloor$ otrzymujemy warunek (1). Warunek (2) możemy przekształcić dalej do równoważnej postaci:

$$\text{istnieją liczby naturalne } k,l \ : \ at = (2k+1)x \text{ oraz } bt = (2l+1)y. \qquad (3)$$

Możemy teraz przystąpić do dowodu równoważności z lematu. Załóżmy najpierw, że promień po strzale w punkt (a,b,c) trafi w krawędź. Wymnażając równości (3), otrzymujemy:

$$at \cdot (2l+1)y \quad = \quad (2k+1)x \cdot bt,$$

więc dzieląc równość stronami przez t i nakładając funkcję *ord* dostajemy:

$$ord(ay(2l+1)) \quad = \quad ord(bx(2k+1))$$

i dalej

$$ord(a) + ord(y) \quad = \quad ord(b) + ord(x)$$

Aby udowodnić implikację w drugą stronę przyjmijmy, że $ord(a) - ord(b) = ord(x) - ord(y)$ i wybierzmy

$$t = 2^{(ord(x)-ord(a))} \cdot NWW\left(\frac{x}{2^{ord(x)}}, \frac{y}{2^{ord(y)}}\right).$$

Wówczas:

$$\frac{at}{x} = \frac{a}{2^{ord(a)}} \frac{NWW\left(\frac{x}{2^{ord(x)}}, \frac{y}{2^{ord(y)}}\right) 2^{ord(x)}}{x},$$

$$\frac{bt}{y} = \frac{b}{2^{ord(b)}} \frac{NWW\left(\frac{x}{2^{ord(x)}}, \frac{y}{2^{ord(y)}}\right) 2^{ord(y)}}{y}.$$

Pokażemy, że tak określone at/x jest nieparzystą liczbą całkowitą. Liczba $\frac{a}{2^{ord(a)}}$ jest oczywiście całkowita i nieparzysta. Spójrzmy teraz na drugi ułamek. Jest on liczbą całkowitą, ponieważ $x = \frac{x}{2^{ord(x)}} \cdot 2^{ord(x)}$ oraz $\frac{x}{2^{ord(x)}} \mid NWW\left(\frac{x}{2^{ord(x)}}, \frac{y}{2^{ord(y)}}\right)$. Pozostaje wykazać, że jest to liczba nieparzysta. Ale to jest oczywiste, ponieważ $2^{ord(x)} \mid x$, a $NWW\left(\frac{x}{2^{ord(x)}}, \frac{y}{2^{ord(y)}}\right)$ jest liczbą nieparzystą.

Analogiczny dowód pozwala wykazać, że także bt/y jest nieparzystą liczbą całkowitą. W ten sposób pokazaliśmy, że punkty (at, bt) i (x,y) są równoważne, a więc promień po t krokach uderzy w krawędź. Równoważność warunków (1) oraz (2) pozwala nam przy tym stwierdzić, że uderzy w krawędź przed powrotem do środka pułapki. ■

Wiosek 3 Można sprawdzić w czasie $O(\log(xyz))$, czy promień po strzale w punkt (a,b,c) trafi w krawędź, zanim wróci do środka pułapki.

Po zastosowaniu opisanych ulepszeń mamy następującą procedurę wyliczania liczby kroków potrzebnych do powrotu do środka pułapki dla strzału w punkt pierwotny (a,b,c).

```
procedure sprawdźStrzał1(a,b,c)
  begin
    if ord(a)-ord(b)=ord(x)-ord(y) then return 0;
    if ord(a)-ord(c)=ord(x)-ord(z) then return 0;
    if ord(b)-ord(c)=ord(y)-ord(z) then return 0;
    return NWW(NWW(2x,a)/a, NWW(2y,b)/b, NWW(2z,c)/c);
  end
```

Procedura *sprawdźStrzał1* działa w czasie $O(\log(xyz))$. Zastosowanie jej w algorytmie `luss1` pozwala nam znaleźć najdłuższą drogę promienia w czasie $O(xyz \cdot \log(xyz))$. Niestety, taki algorytm (nazwijmy go `luss2`) wciąż działa zbyt wolno i przechodzi również tylko pierwszy test.

Odcinanie

Ostatnie usprawnienie algorytmu będzie oparte na dwóch spostrzeżeniach, które wskażą w jakiej kolejności warto przeglądać punkty pierwotne i kiedy można zakończyć poszukiwania.

Rozważmy strzał w punkt (a,b,c) w pułapce o rozmiarach $2x \times 2y \times 2z$. Poprzednio stosowaliśmy ograniczenie $2x \cdot 2y \cdot 2z$ na liczbę kroków T, po których promień wraca do środka pułapki. Istnieje jednak lepsze oszacowanie.

Wiosek 4

$$T \leqslant 2NWW(x,y,z)$$

Dowód Oczywiście $\frac{NWW(2x,a)}{a} \mid 2x$, więc

$$T = NWW\left(\frac{NWW(2x,a)}{a}, \frac{NWW(2y,b)}{b}, \frac{NWW(2z,c)}{c}\right) \leqslant NWW(2x,2y,2z) = 2NWW(x,y,z)$$

■

Takie oszacowanie pozwala nam sformułować warunek, który pozwala wyeliminować niektóre punkty pierwotne.

Wiosek 5 Jeśli po strzale w punkt (a,b,c) promień wraca do środka pułapki w $T = 2NWW(x,y,z)$ krokach, to lepszy wynik (dłuższą drogę promienia) możemy uzyskać tylko dla punktów (a',b',c'), takich że $a'+b'+c' > a+b+c$.

Powyższe dwa wnioski sugerują, że punkty pierwotne należy analizować począwszy od największych wartości sumy ich współrzędnych — tak też robimy w poniższym algorytmie.

```
procedure Lus;
  begin
    D=0;
    Tmax=2NWW(x,y,z);
    for a+b+c = x+y+z to 1 do
      if NWD(a,b,c)=1 then
        t=sprawdźStrzał1(a,b,c);
        if t=Tmax then return (a,b,c);
        if t(a+b+c) > D then
          D = t(a+b+c);
          (aOpt,bOpt,cOpt)=(a,b,c);
    if D > 0 then return (aOpt,bOpt,cOpt)
    else return NULL;
  end
```

Tak ulepszony algorytm przechodzi już wszystkie testy dla zadania. Pozostaje jednak bardzo istotna kwestia. Czy szybkie odnajdowanie właściwego punktu w narożniku pułapki (tam są punkty, dla których suma współrzędnych jest duża) jest dziełem przypadku, wynika z niedoskonałości testów, czy też tak po prostu musi być.

Tak musi być

Czytelnik usatysfakcjonowany posiadaniem szybkiego algorytmu może w tym miejscu zakończyć lekturę. Jeśli jednak interesuje go, dlaczego algorytm działa szybko, to trzeba jeszcze przebrnąć przez poniższy dowód na istnienie szczęścia.

Udowodnimy, że zawsze istnieje strzał, dla którego $T = T_{max}$, w odległości nie większej niż 141 (w metryce miejskiej) od narożnika pułapki. Jest to szacowanie dość grube: w praktyce sprawdzanie kończy się znacznie szybciej — przetestowanie wszystkich przypadków pokazało, że nie istnieje pułapka, w której trzeba strzelić w punkt odległy od narożnika (x,y,z) o więcej niż 7 (nadal w metryce miejskiej). Uzasadnienie rozpoczniemy od wykazania pewnych własności grup liczb.

Definicja 1 Dla liczb całkowitych n i m powiemy, że są one *względnie pierwsze co do 2*, jeśli $NWD(n,m)$ jest potęgą dwójki.

Lemat 4 *Niech* $48 \leqslant x \leqslant 1000$. *W każdym z wypisanych niżej czterech ciągów znajdują się dwie kolejne liczby względnie pierwsze co do 2 z liczbą x.*

(a) $x-1-4i$ *dla* $i = 0,1,\ldots,10$;

(b) $x-2-4i$ *dla* $i = 0,1,\ldots,9$;

(c) $x-3-4i$ *dla* $i = 0,1,\ldots,11$;

(d) $x-4$, $x-8$.

Dowód Dla pierwszych trzech przypadków, pokażemy lemat przez doprowadzenie do sprzeczności. Założymy, że dla każdej pary liczb $x-c-4i$, $x-c-4(i+1)$ z ciągu (a), (b) lub (c) istnieje nieparzysta liczba $d > 1$, taka że

$$d \mid NWD(x, x-c-4i) \quad \text{lub} \quad d \mid NWD(x, x-c-4(i+1)). \tag{4}$$

W dowodzie będziemy korzystać z następującego, oczywistego spostrzeżenia:

$$\text{jeśli } d = NWD(x, x-a), \text{ to } d \mid a, \tag{5}$$

więc dla $d > 1$ i liczby pierwszej a mamy wtedy $a \mid x$.

Pokażmy teraz sprzeczność pomiędzy założeniami lematu i założeniem (4) dla ciągów (a), (b) i (c).

(a) Dla pierwszej pary ciągu: $x-1$ i $x-5$, mamy $NWD(x, x-1) = 1$, więc z założenia (4) wynika, że $NWD(x, x-5) > 1$, a to z własności (5) oznacza iż $5 \mid x$. Następnie rozważmy parę: $x-13, x-17$. Ponownie z założenia (4) i własności (5) otrzymujemy, że $13 \mid x$ lub $17 \mid x$. Po rozważeniu pary: $x-37$, $x-41$, mamy $37 \mid x$ lub $41 \mid x$. Podsumowując, widzimy, że $x \geqslant 5 \cdot 13 \cdot 37 > 1000$, co jest sprzeczne z założeniami lematu.

(b) W tym przypadku rozważamy kolejno pary elementów ciągu: $(x-2, x-6)$, $(x-10, x-14)$, $(x-22, x-26)$ oraz $(x-34, x-38)$. Otrzymujemy, że $3 \mid x$, następnie, że $5 \mid x$ lub $7 \mid x$, potem, że $11 \mid x$ lub $13 \mid x$ i na koniec, że $17 \mid x$ lub $19 \mid x$. Podsumowując widzimy, że $x \geqslant 3 \cdot 5 \cdot 11 \cdot 17 > 1000$, więc doszliśmy do sprzeczności.

(c) Z tego ciągu rozważamy pary:$(x-7, x-11)$, $(x-19, x-23)$, $(x-43, x-47)$. Pozwala to nam zauważyć, że $7 \mid x$ lub $11 \mid x$, następnie $19 \mid x$ lub $23 \mid x$ i na koniec $43 \mid x$ lub $47 \mid x$. Stąd widzimy, że $x \geqslant 7 \cdot 19 \cdot 43 > 1000$, więc także doszliśmy do sprzeczności.

Zauważmy jeszcze, że liczby $x-4$ i $x-8$ są względnie pierwsze z co do 2 z liczbą x wprost z własności (5). To kończy dowód lematu. ■

Wiosek 6 Jeśli $x > 4$ oraz $c \in \{0,1,2,3\}$, to w zbiorze $\{-x, -x+1, \ldots, -1, 1, 2, \ldots, x\}$ istnieją dwie różne liczby a i b spełniające warunki:

- $a = c + 4i$ oraz $b = c + 4j$ dla pewnych całkowitych i, j;
- a i b są względnie pierwsze co do 2 z liczbą x;
- $a = b + 2^k$, dla pewnego $k \geqslant 2$;
- $|a-x| \leqslant 47$ oraz $|b-x| \leqslant 47$.

Dowód Dla $4 < x < 48$ dowód został przeprowadzony przez sprawdzenie wszystkich przypadków (za pomocą odpowiedniego programu). Dla $x \geqslant 48$ możemy zastosować wprost lemat 4. ■

Teraz pokażemy, że nawet w dość silnie ograniczonym zbiorze punktów znajdziemy zawsze punkt pierwotny. W poniższym lemacie będziemy dla punktu $p = (a, b, c)$ używali oznaczenia $NWD(p) = NWD(a, b, c)$.

Lemat 5 *Niech $a,b,c \leqslant 1000$ i niech przynajmniej jedna z tych liczb będzie nieparzysta. Rozważmy zbiory $A = \{a, a-d_a\}$, $B = \{b, b-d_b\}$, $C = \{c, c-d_c\}$, gdzie liczby d_a, d_b, d_c są potęgami dwójki większymi niż 1, takimi że $A \subset (-x,x]$, $B \subset (-y,y]$ oraz $C \subset (-z,z]$. Zdefiniujmy zbiór ośmiu punktów $S = A \times B \times C$. Wśród nich istnieje przynajmniej jeden punkt pierwotny.*

Dowód Załóżmy, że lemat nie jest prawdziwy. Pokażemy, że doprowadza to do sprzeczności.

Bez utraty ogólności możemy przyjąć, że to a jest liczbą nieparzystą. Wówczas także $a - d_a$ jest nieparzysta. Stąd widzimy, że każdy punkt $s \in S$ ma jedną współrzędną nieparzystą i stąd liczba $NWD(s)$ także jest nieparzysta.

Teraz rozważmy dwa różne punkty $s' = (a', b', c')$ i $s'' = (a'', b'', c'')$ ze zbioru S. Pokażemy, że liczby $NWD(s')$ i $NWD(s'')$ są względnie pierwsze. Gdyby istniała liczba $d > 1$, taka że $d \mid NWD(s')$ i $d \mid NWD(s'')$, to mielibyśmy także: $d \mid a' - a''$, $d \mid b' - b''$, $d \mid c' - c''$. Przynajmniej jedna z tych różnic jest niezerowa i jest potęgą dwójki — nie jest możliwe, by dzieliła ją nieparzysta liczba $d > 1$. Stąd mamy, że dla różnych punktów $s', s'' \in S$ zachodzi $NWD(NWD(s'), NWD(s'')) = 1$.

Oznaczmy $D = \{NWD(s) : s \in S\}$. Zbiór D składa sie z ośmiu liczb nieparzystych, które są parami względnie pierwsze. Jeśli nie ma wśród nich jedynek (czyli w S nie ma punktów pierwotnych), to

$$\prod_{d \in D} d \geqslant 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 > 10^6.$$

Zobaczmy teraz, co wynika z powyższego dla liczb a i $a - d_a$. Liczba a występuje jako pierwsza współrzędna czterech punktów ze zbioru S, stąd dzieli ją połowa liczb ze zbioru D. Podobnie jest dla liczby $a - d_a$. Ponieważ liczby a i $a - d_a$ są względnie pierwsze (przyjęliśmy, że a jest nieparzyste, a d_a jest potęgą dwójki), więc element zbioru D nie może być jednocześnie dzielnikiem liczby a i $a - d_a$. Stąd

$$\prod_{d \in D} d \ \mid \ a(a - d_a),$$

więc $a(a - d_a) > 10^6$, co jest sprzeczne za założeniami lematu. ∎

Teraz pozostaje nam wykorzystać udowodnione własności, by wykazać, że blisko narożnika pułapki istnieje punkt pierwotny, w który możemy strzelić uzyskując promień powracający do środka pułapki po T_{max} krokach. Po znalezieniu takiego punktu $s = (a, b, c)$ nie będziemy musieli już rozważać punktów $s' = (a', b', c')$, gdzie $a + b + c \geqslant a' + b' + c'$, bo droga ich promienia na pewno nie będzie dłuższa od $T_{max}(a + b + c)$. Ponieważ $(a + b + c)$ jest wartością stosunkowo dużą, pozwoli to nam istotnie zawęzić obszar poszukiwań.

Twierdzenie 6 *Istnieje punkt pierwotny (a,b,c), taki że $(a,b,c) \in [x-47, x] \times [y-47, y] \times [z-47, z]$ (czyli $a + b + c - (x + y + z) \leqslant 141$) i strzał w ten punkt nie trafia w krawędź pułapki.*

Dowód Dla ustalenia uwagi przyjmijmy, że $ord(x) \geqslant ord(y) \geqslant ord(z)$. Korzystając trzykrotnie z wniosku 6 wybierzmy liczby $a, a - d_a \in [-x, x]$, $b, b - d_b \in [-y, y]$ oraz $c, c - d_c \in [-z, z]$ takie że:

(i) d_a, d_b i d_c są potęgami dwójki nie mniejszymi niż 4;

(ii) $ord(a) = 0$, $ord(b) = 1$ oraz $ord(c) \geqslant 2$;

(iii) liczby a i $a - d_a$ są względnie pierwsze co do 2 z liczbą x, liczby b i $b - d_b$ są względnie pierwsze co do 2 z liczbą y, a liczby c i $c - d_c$ są względnie pierwsze co do 2 z liczbą z;

(iv) $x - a \leqslant 47$ i $x - (a - d_a) \leqslant 47$ oraz $y - b \leqslant 47$ i $y - (b - d_b) \leqslant 47$ oraz $z - c \leqslant 47$ i $z - (c - d_c) \leqslant 47$.

Definiując zbiory $A = \{a, a - d_a\}$, $B = \{b, b - d_b\}$ oraz $C = \{c, c - d_c\}$ z lematu 5 wiemy, że w zbiorze $A \times B \times C$ istnieje punkt pierwotny — oznaczmy go $p = (a', b', c')$. Wówczas liczba $a - d_a$ jest nieparzysta, zachodzi też równość $b - d_b = 2 \ mod \ 4$, a liczba $c - d_c$ jest podzielna przez 4. Dalej mamy: $ord(a) = ord(a - d_a) = 0$, $ord(b) = ord(b - d_b) = 1$, $ord(c) >= 2$, $ord(c - d_c) >= 2$, wobec tego $ord(a') < ord(b') < ord(c')$, czyli

$$
\begin{aligned}
ord(a') - ord(b') \ &< \ 0 \ \leqslant \ ord(x) - ord(y), \\
ord(a') - ord(c') \ &< \ 0 \ \leqslant \ ord(x) - ord(z), \\
ord(b') - ord(c') \ &< \ 0 \ \leqslant \ ord(y) - ord(z).
\end{aligned}
$$

Z lematu 3 widzimy więc, że po strzale w punkt (a', b', c') promień nie trafi w żadną krawędź.

Pozostaje wykazać, że promień przebędzie maksymalną drogę przed powrotem do środka pułapki. Zauważmy, że

$$
\begin{aligned}
T \quad &= \quad NWW\Big(\frac{NWW(2x, a')}{a'}, \frac{NWW(2y, b')}{b'}, \frac{NWW(2z, c')}{c'}\Big) \\
&\geqslant \quad NWW\big(2x, \frac{y}{2^{ord(y)}}, \frac{z}{2^{ord(z)}}\big) = 2NWW(x, y, z) = T_{max},
\end{aligned}
$$

ponieważ a' jest liczbą nieparzystą i $ord(x)$ jest największe spośród $ord(x)$, $ord(y)$ i $ord(z)$.

To kończy dowód twierdzenia.

∎

Implementacja rozwiązania zadania

Rozwiązania poprawne

Zaimplementowano cztery poprawne rozwiązania zadania.

`lus.c`: rozwiązanie wzorcowe działające zgodnie z algorytmem Lus, czyli polegające na przeglądaniu punktów pierwotnych (każdy punkt sprawdzamy w czasie $O(log(xyz))$) począwszy od narożnika pułapki; algorytm kończy działanie, gdy znajdzie punkt, dla którego powrót do środka pułapki następuje po $T = T_{max}$ krokach. Z twierdzenia 6 wiemy, że algorytm zawsze kończy działanie po przejrzeniu stałej liczby punktów. Tak naprawdę nie ma przykładu, dla którego odległość rozwiązania od rogu jest większa niż 7 (choć w dowodzie twierdzenia pokazaliśmy trochę gorsze oszacowanie), więc program dość szybko znajduje rozwiązania.

luss1.c: w programie przeglądamy wszystkie możliwe punkty dla każdego sprawdzając, czy w kolejnych krokach nie trafiliśmy w krawędź. Algorytm działa w czasie $O((xyz)^2)$.

luss2.c: przeglądamy wszystkie możliwe punkty, dla każdego licząc jak program wzorcowy poprawność i drogę strzału — czas działania wynosi $O(xyz\log(xyz))$.

luss3.c: dla każdego punktu sprawdzamy czas powrotu do środka w czasie $O(xyz)$ jak w programie `luss1.c`. Kończymy działanie po znalezieniu punktu, dla którego czas powrotu jest maksymalny, czyli obliczenia wymagają czasu $O(xyz \cdot S)$, gdzie S jest liczbą sprawdzonych punktów.

Jak wcześniej wspomnieliśmy, oprócz rozwiązania wzorcowego, pozostałe programy przechodziły tylko jeden test otrzymując $1/10$ punktów za rozwiązanie.

Błędne rozwiązania

Błędy pojawiające się w rozwiązaniach tego zadania, to przede wszystkim:

- stosowanie prostych heurystyk wyboru najlepszego punktu, na przykład $(x-1, y-1, z-1)$ dla różnych rozmiarów pułapki x, y, z oraz punktu $(x-1, y, z)$, gdy wymiary x i y są równe; heurystyki te nie dają poprawnych rozwiązań;
- niepoprawne rozpoznawanie, czy promień uderzy w krawędź;
- testowanie tylko kilku (ale nie wystarczająco wielu) punktów w narożniku pułapki (w odległości 6).

Uwagi

Zadanie to okazało się najtrudniejszym zadaniem finału — rozwiązał je poprawnie tylko jeden zawodnik przedstawiając algorytm działający podobnie jak program wzorcowy `lus.c`. Testy, choć jak widać z analizy zadania dopuszczały możliwość uzyskania punktów także za rozwiązania nie do końca optymalne, nie zostały pomyślnie rozwiązane przez inne programy zawodników. Także, co w tego typu zadaniach należy uznać za sukces opracowujących testy, nie udało się zdobyć żadnych punktów za pomocą żadnej heurystyki nie dającej poprawnego rozwiązania.

Algorytm wzorcowy pokonuje testy w niezauważalnym czasie. W trakcie opracowywania zadania zaprogramowano także rozwiązania dające poprawną odpowiedź tylko w niektórych przypadkach (sprawdzanie zbyt małej liczby punktów w sąsiedztwie narożnika) bądź działające wystarczająco szybko tylko w niektórych przypadkach (dla pułapek rozmiaru 100 wystarczy szukać optymalnego punktu wśród odległych od narożnika najwyżej o 6).

Testy

Rozwiązania zawodników były sprawdzane na dziesięciu testach. Dodatkowo przygotowano pięć testów dostarczonych zawodnikom w czasie zawodów.

Nazwa	Opis
lus1.in	10 pułapek z zakresu do 100, kilka sześcianów i kilka takich, gdzie odległość rozwiązania od rogu pułapki wynosi 6
lus2.in	wszystkie pułapki o wymiarach z zakresu $[5, 21]$
lus3.in	wszystkie pułapki o wymiarach z zakresu $[991, 1000]$
lus4.in	seria tysiąca dużych losowych pułapek, w tym pułapki, gdzie odległość rozwiązania od rogu pułapki wynosi 7
lus5.in	396 pułapek, każda ma odległość rozwiązania od rogu pułapki wynoszącą 7
lus6.in	wszystkie sześciany mieszczące się w zakresach plus cztery pułapki o odległości rozwiązania od rogu pułapki wynoszącej 7
lus7.in	seria 995 losowych pułapek o dwóch współrzędnych równych plus cztery pułapki o odległości rozwiązania od rogu pułapki wynoszącej 7 plus jakiś sześcian
lus8.in	wszystkie pułapki, gdzie $x \in \{81, 82, \ldots, 90\}$, $y \in \{71, 72, \ldots, 80\}$, $z \in \{41, 42, \ldots, 50\}$
lus9.in	seria 30 losowych pułapek o wielkości do 100, w tym jedna, dla której rozwiązanie jest w odległości 6 od rogu pułapki
lus10.in	seria 30 pułapek o wielkości do 100, gdzie odległość rozwiązania od rogu pułapki wynosi 6

XVI Międzynarodowa
Olimpiada Informatyczna

Ateny, Grecja 2004