Marian M. Kędzierski, Jakub Radoszewski — Treść zadania, Opracowanie
Tomasz Kulczyński — Program

Dostępna pamięć: 64 MB. *OI, Etap I, 20.10–17.11.2008*

Straż pożarna

W stolicy Bajtocji, Bajtawie, ulice mają bardzo regularny układ. Wszystkie biegną albo z północy na południe, albo z zachodu na wschód. Łatwo zauważyć, że każda ulica z północy na południe przecina każdą ulicę z zachodu na wschód w dokładnie jednym miejscu. Ponadto, wzdłuż każdej ulicy kolejne skrzyżowania są odległe o dokładnie 1 km.

W Bajtawie jest z zabytkowych budynków, z których każdy znajduje się przy jednym ze skrzyżowań. Radzie Miejskiej bardzo zależy na ochronie tych unikalnych zabytków, dlatego postanowiono wybudować w mieście dwa duże posterunki straży pożarnej. Każdy z zabytków będzie chroniony przez posterunek jemu najbliższy; w przypadku równych odległości od każdego z posterunków, budynek będzie pod ochroną ich obu.

Zabudowa w Bajtawie jest bardzo gęsta. Nie należy więc patrzeć na odległość do zabytków w linii prostej. Zamiast tego, jako odległość od posterunku do zabytku należy przyjąć długość najkrótszej trasy biegnącej ulicami.

Rada Miejska przygotowała kilka projektów lokalizacji posterunków straży. Zostałeś poproszony o wyznaczenie, dla każdego z nich, liczby zabytków chronionych: tylko przez pierwszy posterunek, tylko przez drugi posterunek oraz przez oba posterunki.

Wejście

W pierwszym wierszu standardowego wejścia znajdują się cztery liczby całkowite n, m, z oraz p ($1 \leqslant n,m \leqslant 1\,000\,000\,000$, $1 \leqslant z,p \leqslant 100\,000$) pooddzielane pojedynczymi odstępami, oznaczające odpowiednio: liczbę ulic biegnących z północy na południe, liczbę ulic biegnących z zachodu na wschód, liczbę zabytkowych budynków w Bajtawie oraz liczbę projektów zaproponowanych przez Radę Miejską. Ulice biegnące z północy na południe są ponumerowane od 1 do n, w kierunku z zachodu na wschód. Ulice biegnące z zachodu na wschód są ponumerowane od 1 do m, w kierunku z północy na południe. Skrzyżowaniu x-tej ulicy biegnącej z północy na południe z y-tą ulicą biegnącą z zachodu na wschód dla uproszczenia przypisujemy współrzędne (x,y).

W każdym z kolejnych z wierszy znajdują się dwie liczby całkowite x_i oraz y_i ($1 \leqslant x_i \leqslant n$, $1 \leqslant y_i \leqslant m$) oddzielone pojedynczym odstępem i oznaczające współrzędne i-tego zabytku. Żadna para zabytków nie znajduje się przy tym samym skrzyżowaniu.

Każdy z kolejnych p wierszy zawiera jedną propozycję Rady Miejskiej — cztery liczby całkowite $x_{j,1}$, $y_{j,1}$, $x_{j,2}$, $y_{j,2}$ pooddzielane pojedynczymi odstępami, $1 \leqslant x_{j,1},x_{j,2} \leqslant n$, $1 \leqslant y_{j,1},y_{j,2} \leqslant m$, $(x_{j,1},y_{j,1}) \neq (x_{j,2},y_{j,2})$. Współrzędne $(x_{j,1},y_{j,1})$ oraz $(x_{j,2},y_{j,2})$ opisują skrzyżowania, przy których mają być umiejscowione posterunki straży zgodnie z j-tą propozycją ($1 \leqslant j \leqslant p$).

Wyjście

Twój program powinien wypisać na standardowe wyjście p wierszy. W j-tym wierszu powinny się znaleźć trzy liczby całkowite, oznaczające: liczbę zabytków chronionych tylko przez pierwszy

posterunek z j-tej propozycji Rady Miejskiej, liczbę zabytków chronionych tylko przez drugi posterunek oraz liczbę budynków chronionych przez oba posterunki. Liczby te powinny być oddzielone pojedynczymi odstępami.

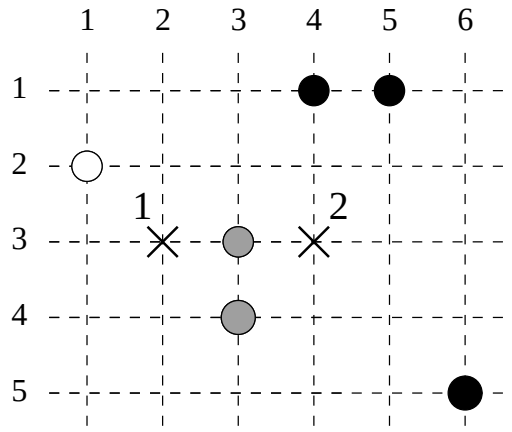
Przykład

Dla danych wejściowych:

```
6 5 6 1
1 2
6 5
5 1
3 3
3 4
4 1
2 3 4 3
```

poprawnym wynikiem jest:

```
1 3 2
```



Na rysunku linie przerywane przedstawiają ulice, kółka — lokalizacje zabytków, a krzyżyki — proponowane lokalizacje posterunków straży pożarnej. Białe kółko przedstawia zabytek chroniony przez pierwszy posterunek, czarne kółka — przez drugi posterunek, natomiast szare kółka — przez oba posterunki.

Rozwiązanie

Wstęp

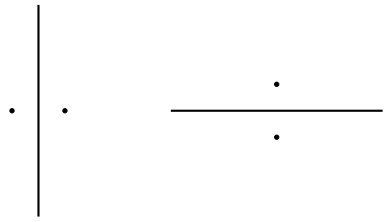
Sformułujmy treść naszego zadania w języku geometrii. Na płaszczyźnie znajduje się z punktów $P_1, P_2, \ldots, P_z$ reprezentujących zabytki Bajtawy. Dodatkowo, danych jest p par punktów S_j, T_j będących propozycjami lokalizacji posterunków straży pożarnej. Naszym zadaniem jest wyznaczenie, dla każdej propozycji, liczby zabytków położonych: bliżej S_j niż T_j (obszar A), bliżej T_j niż S_j (obszar B) oraz w jednakowej odległości do S_j i T_j (obszar R). Zgodnie z treścią zadania, odległość między punktami (x_1, y_1) i (x_2, y_2) jest mierzona w tzw. *metryce miejskiej* (nazywanej też metryką Manhattan), czyli:

$$d\big((x_1, y_1), (x_2, y_2)\big) = |x_1 - x_2| + |y_1 - y_2|.$$

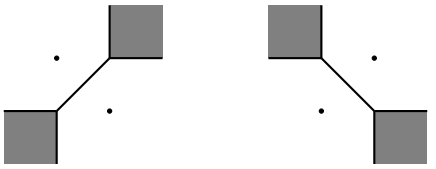
Tak sformułowane zadanie bardzo łatwo rozwiązać siłowo, po prostu wyznaczając wszystkie potrzebne odległości i porównując je. W ten sposób uzyskuje się rozwiązanie o złożoności czasowej $O(p \cdot z)$, czyli mało efektywne, biorąc pod uwagę ograniczenia z zadania. Jego implementacje można znaleźć w plikach `strs1.cpp` i `strs2.pas`; zdobywało ono na zawodach 20% punktów.

Podział na obszary

Okazuje się, że dobrym krokiem ku skonstruowaniu efektywniejszego rozwiązania jest dokładniejsze przyjrzenie się kształtom zdefiniowanych powyżej obszarów A, B i R. Jak łatwo się domyślić, są one istotnie zależne od wzajemnego położenia posterunków $S_j = (x_1, y_1)$ i $T_j = (x_2, y_2)$. W sumie można wyliczyć osiem różnych przypadków, zilustrowanych poniżej. Na rysunkach czarne kropki oznaczają położenia posterunków, natomiast linie oraz zacieniowane prostokąty reprezentują obszar remisowy R i odgraniczają od siebie obszary typu A oraz B.



Przypadek 1. $x_1 = x_2$ lub $y_1 = y_2$.



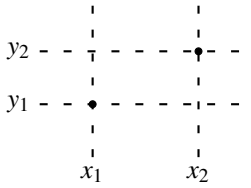
Przypadek 2. $|x_1 - x_2| = |y_1 - y_2|$.

Przypadek 3. Pozostałe konfiguracje.

Wnikliwy Czytelnik zapewne zdążył już sobie postawić pytanie, skąd taki akurat podział na przypadki i w jaki sposób zostały skonstruowane poszczególne diagramy[1]. Istnieją różne możliwe odpowiedzi na to pytanie. Jedna z metod polega na wybieraniu „do skutku" różnych wzajemnych położeń par posterunków i rysowaniu linii podziału na obszary, na przykład poprzez przyglądanie się punktom o współrzędnych całkowitoliczbowych. W ten sposób jednak łatwo o przeoczenie — na przykład w przypadku $|x_1 - x_2| = |y_1 - y_2|$ można nie zauważyć dużych obszarów remisowych.

[1] Rozważane przez nas diagramy stanowią szczególny przypadek tak zwanych *diagramów Voronoi*. W ogólnej wersji dana jest pewna liczba punktów, a naszym zadaniem jest podzielenie płaszczyzny na obszary najbliższe poszczególnym z tych punktów, zazwyczaj w standardowej metryce euklidesowej. Więcej o diagramach Voronoi można poczytać np. w książkach [36] i [39].

Istnieje także bardziej metodyczne podejście do tego problemu. Najłatwiej zidentyfikować i rozpatrzyć przypadki typu 1. We wszystkich pozostałych mamy $x_1 \neq x_2$ i $y_1 \neq y_2$, co pozwala wydzielić dwie możliwe sytuacje: $x_1 < x_2$ i $y_1 < y_2$ oraz $x_1 < x_2$ i $y_1 > y_2$ (pozostałe możliwości są takie same z dokładnością do zamiany posterunków miejscami). Dla każdej z nich możemy podzielić płaszczyznę na 9 obszarów, odpowiadających produktom kartezjańskim przedziałów $[-\infty, x_1]$, $[x_1, x_2]$ i $[x_2, \infty]$ oraz $[-\infty, y_1]$, $[y_1, y_2]$ i $[y_2, \infty]$, co widać na rys. 1.



Rys. 1: Podział płaszczyzny na 9 obszarów w przypadku $x_1 < x_2$, $y_1 < y_2$.

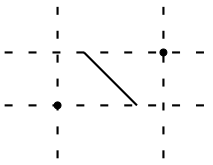
Jeżeli teraz punkt $P = (x, y)$ należy do ustalonego spośród tych obszarów, to każdą z nierówności: $d(P, S_j) < d(P, T_j)$, $d(P, S_j) > d(P, T_j)$ oraz równanie $d(P, S_j) = d(P, T_j)$ można już łatwo rozwiązać. Dla przykładu, jeżeli P należy do środkowego obszaru $[x_1, x_2] \times [y_1, y_2]$, to pierwsza ze wspomnianych nierówności ma postać

$$x - x_1 + y - y_1 < x_2 - x + y_2 - y,$$

czyli

$$x + y < \frac{x_1 + y_1 + x_2 + y_2}{2}, \tag{1}$$

co odpowiada środkowej ukośnej kresce na każdym z diagramów w przypadkach 2 i 3:



Rys. 2: Pierwsza kreska diagramu.

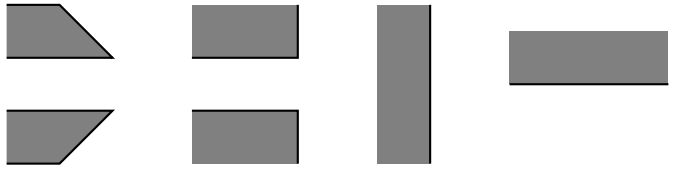
Kontynuując to postępowanie, można wypisać wspomniane nierówności dla poszczególnych obszarów — czasem okazują się one trywialne, czyli zawsze prawdziwe lub zawsze fałszywe (jak na przykład w przypadku lewego dolnego i prawego górnego obszaru na rys. 2), a czasem definiują pewne zależności, które po jednej stronie mają stałą, a po drugiej x, y, $x + y$ lub $x - y$. Dokładniejsza analiza tych nierówności pozwala wykryć, że ich postać zależy już tylko od tego, jak wyrażenie $|x_1 - x_2|$ ma się do wyrażenia $|y_1 - y_2|$, co prowadzi do takiej klasyfikacji przypadków 2 i 3, jak zaprezentowana powyżej.

Podział na podobszary

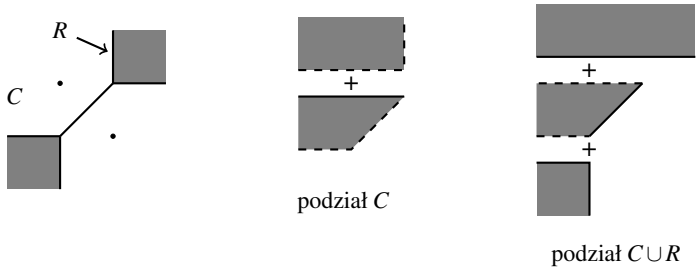
Jako że w zadaniu musimy wyznaczać liczby zabytków P_i, które należą do obszarów A, B, R, to powinno nam zależeć na tym, żeby kształty rozważanych obszarów były możliwie proste i jednolite. Ponieważ diagramy przypadków 1–3 mają stosunkowo urozmaicony kształt, podzielimy poszczególne obszary na mniejsze, regularniejsze części. Im sprytniej to zrobimy, tym łatwiejsza będzie dalsza część rozwiązania.

Zacznijmy od tego, że nigdy nie musimy zliczać zabytków we wszystkich trzech obszarach A, B, R, gdyż zawsze liczby te sumują się do z. W szczególności, możemy pominąć jeden z obszarów A, B. Dla ujednolicenia, do zliczania wybierzemy zawsze ten obszar, który dotyczy górnego lewego spośród posterunków (tzn. lewego, a w razie remisu górnego) — oznaczmy ten obszar przez C. Dodatkowo, zamiast zliczać zabytki w ramach R, wykonamy to dla obszaru $C \cup R$, czyli dla sumy zbiorów C i R. Jest to korzystne, gdyż obszar R ma zazwyczaj kształt zupełnie nieprzypominający C, natomiast kształty C i $C \cup R$ są stosunkowo zbliżone.

Jak teraz łatwo sprawdzić, każdy z obszarów C, $C \cup R$ w każdym z naszych ośmiu przypadków można rozłożyć na stałą liczbę fragmentów sześciu rodzajów przedstawionych na rys. 3. Przykład takiego rozkładu dla lewego podprzypadku przypadku 2 obrazuje rys. 4.



Rys. 3: Podstawowe klocki budulcowe obszarów C i $C \cup R$, od lewej: jednostronnie nieskończone trapezy, dwustronnie nieskończone prostokąty i półpłaszczyzny: pionowa i pozioma. Czarne kreski na brzegach reprezentują fragmenty brzegu, które mogą należeć do figury bądź nie (na każdym prostym fragmencie obwodu charakterystyka przynależności wszystkich punktów będzie jednak zawsze taka sama).

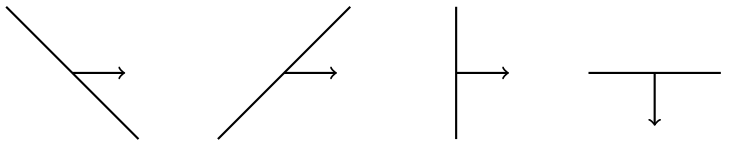


Rys. 4: Rozkład obszarów C oraz $C \cup R$ w drugim przypadku na klocki budulcowe z rys. 3. Linie ciągłe obrazują brzegi zaliczające się do figury, linie przerywane — brzegi otwarte, wreszcie brak linii reprezentuje nieskończony wymiar w danym kierunku.

Zamiatanie

Kluczem do dokończenia naszego rozwiązania jest odstąpienie od podejścia siłowego i rozważanie wszystkich zapytań naraz. Mamy wówczas układ posterunków P_i oraz pewien zbiór obszarów O_j takich jak na rys. 3, a naszym celem jest wyznaczenie liczby posterunków w każdym z tych obszarów. Jeżeli to zrobimy, to na podstawie dotychczasowych rozważań można będzie odtworzyć wyniki dla poszczególnych zapytań.

Przy tak postawionym problemie do rozwiązania możemy wykorzystać klasyczną technikę *zamiatania*[2], która polega na systematycznym (czyli w jakiejś ustalonej kolejności) przejrzeniu płaszczyzny i kolejnym obsługiwaniu napotykanych zdarzeń. W naszym przypadku zdarzeniami są: dodanie punktu P_i oraz zapytanie o liczbę punktów w obszarze O_j. Aby ustalić, jaki konkretnie rodzaj zamiatania jest nam potrzebny, zauważmy, że każdy z obszarów O_j może zostać opisany za pomocą co najwyżej dwóch prostych nierówności na x, y, $x+y$ lub $x-y$, z czego dopuszczamy, aby co najwyżej jedna była podwójna, czyli narzucająca zarówno ograniczenie górne, jak i dolne. To może stanowić wskazówkę, że będzie nam potrzebne nie jedno, ale cztery osobne zamiatania, w których zdarzenia będą uporządkowane odpowiednio po x, y, $x+y$ i $x-y$ (patrz rys. 5).



Rys. 5: Kierunki zamiatania potrzebne do obsłużenia obszarów z rys. 3.

Obszary będące nieskończonymi trapezami będziemy rozważać w zamiataniach, podczas których miotła przemieszcza się w kolejności rosnącej względem odpowiednio $x+y$ (górny obszar na rys. 3) i $x-y$ (dolny). Obsługa zapytania odpowiadającego danemu obszarowi nastąpi w momencie napotkania ukośnego fragmentu brzegu obszaru i polegać będzie na zliczeniu już napotkanych punktów P_i, których współrzędna y należy do przedziału zadanego przez dolny i górny brzeg obszaru. Aby efektywnie odpowiadać na takie zapytania, z miotłą związemy strukturę danych zwaną *drzewem przedziałowym*[3]. Umożliwia ono wykonywanie operacji dodania punktu o danej rzędnej i zapytania o liczbę dodanych punktów o rzędnej z zadanego przedziału w złożoności czasowej $O(\log M)$, gdzie M jest rozważanym zakresem rzędnych. Analogicznie można rozpatrzyć obszary będące nieskończonymi prostokątami i półpłaszczyzny pionowe (tym razem zamiatając w kolejności wzrastających odciętych) oraz półpłaszczyzny poziome (zamiatanie w kolejności malejących rzędnych, czyli „w dół"). Zauważmy, że w drugim z tych przypadków miotła zawierająca same rzędne nie bardzo ma

[2] O zamiataniu lub, jak kto woli, wymiataniu można poczytać we wspomnianych już książkach o geometrii obliczeniowej [36] oraz [39], a także posłuchać na stronie `http://was.zaa.mimuw.edu.pl`. Technika ta była wykorzystywana w rozwiązaniach wielu zadań olimpijskich, jak np. Trójkąty z XV Olimpiady Informatycznej [15] lub Gazociągi z XIV Olimpiady Informatycznej [14].

[3] Więcej o drzewach przedziałowych można przeczytać np. w opisie rozwiązania zadania Tetris 3D z książeczki XIII Olimpiady Informatycznej [13], albo poczytać i posłuchać na stronie `http://was.zaa.mimuw.edu.pl`

sens, jednakże wówczas i tak wszystkie zapytania będą dotyczyły zliczania wszystkich już napotkanych punktów, więc jest to bez znaczenia.

Do dokończenia rozwiązania pozostało nam jeszcze kilka istotnych szczegółów. Przede wszystkim nie wspomnieliśmy jeszcze, że drzewo przedziałowe odpowiadające zakresowi współrzędnej o rozmiarze M ma złożoność pamięciową $O(M)$. To stanowi w naszym przypadku istotny problem, gdyż górne ograniczenia na wartości współrzędnych w zadaniu (tj. n oraz m) są bardzo duże. W takim przypadku należy jednak zastosować jedną ze standardowych technik (także opisanych na stronie `http://was.zaa.mimuw.edu.pl`), na przykład przenumerowanie współrzędnych. Dokładniej, ponieważ przed rozpoczęciem zamiatania z góry możemy przewidzieć wszystkie wartości rzędnych, które wystąpią przy wstawianiu oraz w zapytaniach, więc wystarczy prenumerować te faktycznie dla nas istotne wartości kolejnymi liczbami naturalnymi $0, 1, 2, \ldots$ Samo prenumerowanie wymaga zaledwie jednego sortowania — koszt czasowy $O\big((p+z)\log(p+z)\big)$ — natomiast po jego wykonaniu pozostaje nam już tylko $M = O(p+z)$ istotnych rzędnych — korzystamy w tym miejscu z tego, że każdy z obszarów C, $C \cup R$ rozkłada się na stałą liczbę klocków budulcowych z rys. 3. Innym rozwiązaniem może być użycie dynamicznej (opartej o wskaźniki) wersji drzewa przedziałowego (również opisanej na wspomnianej stronie).

Kolejna trudność wiąże się z otwartością i domkniętością poszczególnych brzegów figur z rys. 3. Jak dotychczas kompletnie to ignorowaliśmy, ale już przykład z rys. 4 pokazuje, że z przynależnością brzegów do obszarów budulcowych może być w rozkładach bardzo różnie. Właściwą metodę radzenia sobie z tym problemem opiszemy znów na przykładzie nieskończonych trapezów. Otwartość bądź domkniętość górnych i dolnych brzegów łatwo obsłużyć, po prostu zadając zapytania do drzewa przedziałowego zawierające przedziały otwarte tudzież domknięte z odpowiednich stron. Natomiast ukośny fragment brzegu musi być rozpatrywany w zupełnie innym miejscu algorytmu, a mianowicie podczas sortowania zdarzeń względem $x+y$ (odpowiednio $x-y$). W tym celu należy podczas sortowania odpowiednio rozstrzygać remisy między punktami P_i a obszarami O_j: w przypadku takiego remisu pierwszeństwo mają obszary o otwartym ukośnym fragmencie brzegu (w dowolnej już kolejności między sobą), dalej następują wszystkie posterunki P_i, a na końcu zostają umieszczone obszary o brzegu domkniętym.

Podsumowanie

Rozwiązanie wzorcowe składa się z następujących kroków:

1. Podział wszystkich zapytań na przypadki 1–3 i dalsze podprzypadki.
2. Podział każdego z fragmentów C oraz $C \cup R$ na obszary budulcowe.
3. Przenumerowanie rzędnych punktów P_i oraz wyznaczających brzegi obszarów O_j za pomocą sortowania i scalenia posortowanych list (opcjonalnie: wyszukiwania binarnego zamiast scalania).
4. Cztery zamiatania:
   (a) sortowanie punktów i odpowiednich obszarów względem, odpowiednio, x, y, $x+y$ bądź $x-y$ (tutaj y reprezentuje współrzędną nieprzenumerowaną);

(b) obsługa kolejnych zdarzeń w posortowanej kolejności z wykorzystaniem drzewa przedziałowego względem rzędnych.

Pierwsze dwa kroki można zrealizować w złożoności czasowej $O(z)$, natomiast dwa następne w koszcie czasowym $O\big((p+z)\log(p+z)\big)$, zdeterminowanym przez sortowania bądź sekwencje operacji na drzewach przedziałowych. Złożoność pamięciowa to $O(p+z)$. Gdybyśmy użyli dynamicznego drzewa przedziałowego zamiast przenumerowania rzędnych, złożoność pamięciowa wyniosłaby $O((p+z)\log M)$, co przy odpowiedniej implementacji również pozwala zmieścić się w limicie 64 MB.

Właściwie wszystkie rozwiązania jurorskie poza tym opisanym we wstępie opierają się na powyższym schemacie. Poprawne implementacje można znaleźć w plikach: `str.cpp`, `str1.pas`, `str2.java` i `str3.cpp`. W pliku `strb1.cpp` znajduje się rozwiązanie błędne, które w niewłaściwy sposób generowało diagramy w przypadku 2 (wspominaliśmy już o tym błędzie) — uzyskiwało ono 30% punktów. Z kolei plik `strb2.cpp` zawiera implementację rozwiązania, które uzyskiwało 70% punktów wskutek błędów przepełnienia typu całkowitego 32-bitowego — zauważmy, że współrzędne w zadaniu ograniczone są z góry przez 10^9, a obliczenie ograniczeń obszarów może wymagać np. czterech dodawań takich liczb, patrz nierówność (1). Dodajmy także, że mniej umiejętny podział obszarów A, B, R na podobszary mógłby spowodować konieczność wykonania większej liczby zamiatań (sześciu, ośmiu...) — implementacje takich rozwiązań jurorzy pozostawili jednak zawodnikom.

Testy

Rozwiązania zawodników były sprawdzane na następujących dziesięciu testach:

Nazwa	z	p	max(n, m)	Opis
str1.in	3	100	10	mały test losowy bez przypadku 2
str2.in	101	513	10^6	mały test losowy bez przypadku 2
str3.in	9 000	83 810	100	pełna kratownica zabytków 100×90
str4.in	5 213	90 213	$\approx 10^7$	średni test losowy
str5.in	28 000	30 000	$\approx 10^6$	średni test losowy
str6.in	63 453	52 342	$\approx 2 \cdot 10^7$	średni test losowy
str7.in	70 013	57 000	$\approx 10^7$	średni test losowy
str8.in	80 000	80 000	$\approx 10^8$	duży test losowy
str9.in	99 876	95 123	$\approx 10^9$	duży test losowy bez przypadku 2
str10.in	100 000	100 000	10^9	maksymalny test losowy

Zawody II stopnia

opracowania zadań