Jakub Wojtaszczyk — Treść zadania, Opracowanie
Adam Polak — Program

Dostępna pamięć: 128 MB. *OI, etap I, 15.10–12.11.2012*

Gobeliny

W Bajtockim Muzeum Sztuk Pięknych rozpoczyna się wystawa gobelinów. Główna sala wystawowa ma, patrząc z góry, kształt wielokąta (niekoniecznie wypukłego). Na każdej ze ścian sali powieszono jeden gobelin. Każdy gobelin zajmuje dokładnie całą powierzchnię ściany.

Do sali wstawiono lampę, która ma oświetlać wystawę. Lampa świeci równomiernie we wszystkich kierunkach. Wiadomo, że niektóre z gobelinów muszą być dobrze oświetlone, a niektórych – wręcz przeciwnie – nie można wystawiać na ostre światło.

Bajtazar, kustosz muzeum, zaczął przesuwać lampę po sali, ale nie udało mu się jej ustawić tak, żeby był zadowolony. Bajtazar jest przerażony – obawia się, że będzie trzeba przewieszać gobeliny (co wymaga dużo pracy), a do otwarcia wystawy zostało bardzo niewiele czasu. Może zdołasz mu pomóc i podpowiesz, czy jego wysiłki w ogóle mają sens?

Twoim zadaniem jest rozstrzygnięcie, czy istnieje położenie lampy spełniające poniższe warunki:

- *każda ściana musi być albo oświetlona w całości, albo zaciemniona w całości, w zależności od wymagań dotyczących danego gobelinu; natomiast nie może istnieć ściana, na której część pada światło, a na część nie;*
- *jeżeli lampa znajduje się dokładnie w linii ściany, to jej nie oświetla;*
- *lampy nie można wyłączyć ani zabrać z sali; musi być włączona i znajdować się wewnątrz sali (w szczególności nie może znajdować się na żadnej ze ścian ani w rogu sali).*

Wejście

W pierwszym wierszu standardowego wejścia znajduje się jedna liczba całkowita t ($1 \leqslant t \leqslant 20$), oznaczająca liczbę zestawów danych. W kolejnych wierszach znajdują się opisy zestawów danych.

W pierwszym wierszu opisu znajduje się jedna liczba całkowita n ($3 \leqslant n \leqslant 1000$), oznaczająca liczbę ścian sali wystawowej. W kolejnych n wierszach opisany jest kształt sali. W każdym z tych wierszy znajdują się dwie liczby całkowite x_i i y_i oddzielone pojedynczym odstępem ($-30\,000 \leqslant x_i, y_i \leqslant 30\,000$ dla $i = 1, 2, \ldots, n$) – są to współrzędne narożnika sali, czyli wierzchołka wielokąta opisującego kształt sali. Wierzchołki są podane w kolejności zgodnej z kierunkiem ruchu wskazówek zegara.

W kolejnych n wierszach opisane są wymagania dotyczące gobelinów wiszących na ścianach. W każdym z tych wierszy znajduje się jedna litera `S` *lub* `C`*, oznaczająca odpowiednio, że ściana ma być oświetlona lub zaciemniona. Litera znajdująca się w i-tym z tych wierszy (dla $1 \leqslant i \leqslant n-1$) dotyczy ściany łączącej wierzchołki i-ty oraz $(i+1)$-szy. Litera znajdująca się w ostatnim z tych wierszy dotyczy ściany łączącej ostatni wierzchołek z pierwszym.*

Wielokąt opisujący kształt sali nie ma samoprzecięć, tzn. poza sąsiednimi bokami, które siłą rzeczy mają wspólny koniec, żadne dwa boki wielokąta nie mają punktów wspólnych. Żadne trzy wierzchołki wielokąta nie są współliniowe.

W testach wartych łącznie 40% punktów zachodzi dodatkowy warunek $n \leqslant 20$. Dodatkowo, w testach wartych łącznie 10% punktów wszystkie ściany mają być oświetlone.

Wyjście

Dla każdego zestawu danych Twój program powinien wypisać na standardowe wyjście jeden wiersz zawierający jedno słowo: `TAK` *– jeżeli da się ustawić lampę zgodnie z podanymi warunkami, lub* `NIE` *– w przeciwnym przypadku.*

Przykład

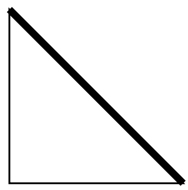
Na rysunkach pogrubione boki oznaczają ściany, które mają być zaciemnione, a pozostałe boki – ściany, które mają być oświetlone. Rysunek do drugiego przykładu przedstawia poprawne położenie lampy.

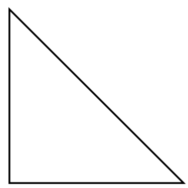
Dla danych wejściowych:

```
2
3
0 0
0 1
1 0
S
C
S
3
0 0
0 1
1 0
S
S
S
```

poprawnym wynikiem jest:

```
NIE
TAK
```



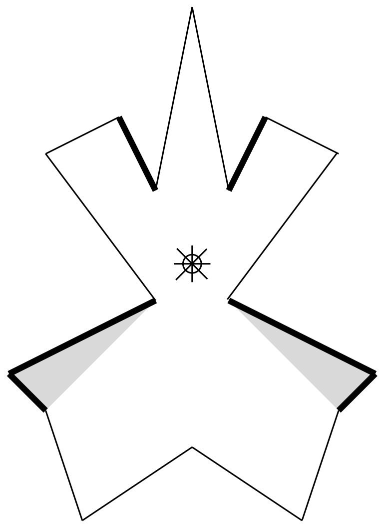


natomiast dla danych:

```
1
16
5 -3
4 -4
3 -7
0 -5
-3 -7
-4 -4
-5 -3
-1 -1
-4 3
-2 4
-1 2
0 7
1 2
2 4
4 3
1 -1
C
S
S
S
S
C
C
S
S
C
S
S
C
S
S
C
```

poprawnym wynikiem jest:

```
TAK
```



Rozwiązanie

Jasne ściany

Zadanie jest niełatwe, więc zacznijmy od rozpatrzenia jego istotnie łatwiejszej wersji – spróbujmy wpierw rozważyć przypadek, w którym kustosz chciałby, żeby wszystkie ściany były oświetlone.

Potrzebujemy zatem znaleźć punkt wewnątrz wielokąta, z którego będzie widać wszystkie ściany. Prostym warunkiem, który taki punkt musi spełniać, jest to, że musi znajdować się „po wewnętrznej stronie" każdej ze ścian. By wyrazić to dokładnej, rozważmy dowolną ścianę wielokąta i przeprowadźmy przez nią linię prostą. W pobliżu ściany, po jednej stronie tej prostej znajduje się wnętrze muzeum, a po drugiej – zewnętrze; półpłaszczyznę po stronie wnętrza nazwiemy „wewnętrzną". Nasz warunek mówi, że dla każdej ściany lampa musi znajdować się po wewnętrznej stronie prostej zawierającej tę ścianę (nie może też znajdować się na samej prostej). Łatwo przekonać się, że jest to warunek konieczny.

Zaczniemy od pokazania, że ten warunek jest wystarczający, by lampa oświetliła wszystkie ściany, a potem zastanowimy się, jak ten warunek sprawdzić. Załóżmy, że lampa jest położona po wewnętrznej stronie każdej ściany, ale pewna ściana nie jest całkowicie oświetlona. Oznacza to, że promień światła idący od lampy do pewnego punktu tej ściany przecina zewnętrze muzeum. Jako że tuż przy ścianie promień znajduje się we wnętrzu muzeum, to gdzieś musi przechodzić z zewnętrza na wnętrze – a to oznacza, że przecina pewną ścianę „od zewnątrz", co nie może się zdarzyć (bo wtedy lampa leżałaby po zewnętrznej stronie tej ściany). Doszliśmy do sprzeczności, zatem każda ściana jest oświetlona. Ten sam argument pokazuje, że lampa znajduje się wewnątrz wielokąta.

Warto wiedzieć, że wielokąt, w którym da się z jednego punktu oświetlić wszystkie ściany, nazywa się *wielokątem gwiaździstym*, zaś zbiór punktów, z których da się wielokąt oświetlić (czyli przecięcie opisanych wyżej półpłaszczyzn), nazywa się jego *jądrem*.

Jasne ściany – algorytm

Jak jednak sprawdzić, czy wielokąt jest gwiaździsty? Najprościej będzie spróbować wyznaczyć jego jądro. Istnieje szereg algorytmów (opierających się na przecinaniu półpłaszczyzn) pozwalających wyznaczyć jądro (a zatem, w szczególności, stwierdzić, czy jest niepuste) w czasie $O(n \log n)$, mają one jednak tę wadę, że są nietrywialne w implementacji – a my przecież rozpatrujemy dopiero prosty przypadek naszego zadania. Na pomoc przyjdzie nam ograniczenie na liczbę boków naszego wielokąta (a zatem na liczbę półpłaszczyzn, które przecinamy) – jest ich co najwyżej 1000, a zatem stać nas na rozwiązanie w czasie $O(n^2)$.

Takie rozwiązanie może wyglądać następująco. Zauważmy, że jeśli jądro jest niepuste, to jest wielokątem wypukłym (bo jest przecięciem półpłaszczyzn, które same są wypukłe), a jego boki są zawarte w prostych wyznaczających półpłaszczyzny (czyli w prostych zawierających boki naszego wielokąta). Zatem możemy spróbować znaleźć bok jądra – bierzemy kolejno każdą z prostych i patrzymy na jej przecięcia z wszystkimi pozostałymi półpłaszczyznami. Te przecięcia są oczywiście półprostymi (ew. prostymi lub zbiorami pustymi, dla półpłaszczyzn wyznaczanych przez proste równoległe do naszej), a więc bardzo łatwo sprawdzić, czy ich przecięcie zawiera jakiś odcinek. Jeśli nie, to na tej prostej nie ma boku jądra, a jeśli tak – to znaleźliśmy bok (a zatem jądro jest niepuste). Nieprzyjemną sytuacją do rozważenia byłoby, gdyby dwie półpłaszczyzny były wyznaczone przez tę samą prostą, szczęśliwie w treści zadania mamy zagwarantowane, że żadne trzy wierzchołki wielokąta nie są współliniowe, co wyklucza ten przypadek.

Rozwiązania, które rozpatrywały tylko jasne ściany, uzyskiwały 10% punktów za to zadanie. Przykładowa implementacja znajduje się w pliku `gobb1.cpp`.

Ciemne ściany

Zacznijmy od zauważenia, że jeśli wszystkie ściany mają być ciemne, to lampy ustawić się nie da – światło gdzieś musi się podziać.

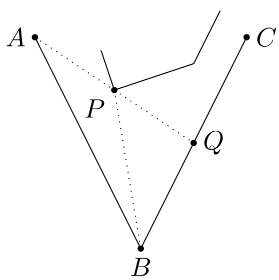
Byłoby pięknie, gdyby zbiór możliwych ustawień lampy wyznaczany przez ciemną ścianę też był półpłaszczyzną. Niestety, nie jest – lampa może ściany nie oświetlać dlatego, że znajduje się „za nią” (czyli w „zewnętrznej” półpłaszczyźnie), ale może też być zasłonięta przez jakąś inną ścianę; przykłady obydwu tych sytuacji widzimy w teście przykładowym.

Przyjrzenie się testowi przykładowemu może nam nasunąć jednak pewne przemyślenie. Otóż jeśli mamy obok siebie dwie sąsiadujące ciemne ściany, które tworzą kąt wypukły (czyli w punkcie ich styku jest skręt do środka wielokąta), to – jak widać na przykładowym obrazku – zacieniony jest cały trójkąt, którego dwoma bokami są te dwie ściany. Jest to prawdą przynajmniej wtedy, gdy w tym trójkącie nie ma żadnej innej ściany – faktycznie, gdyby docierał tam jakikolwiek promień światła, to musiałby skądś przyjść i gdzieś wyjść – a prosta nie może dwa razy przecinać tego samego boku trójkąta. Zatem, w tej szczególnej sytuacji, dwie zacienione ściany możemy zastąpić jedną, będącą trzecim bokiem rozważanego trójkąta, i ta ściana również musi być zacieniona.

To spostrzeżenie może budzić naszą nadzieję, że coś da się z zadaniem zrobić, albowiem pozwoliło nam zredukować zadanie do mniejszego – takiego, w którym mamy o jedną ciemną ścianę mniej. Spróbujmy sobie zatem poradzić z podobnymi przypadkami – gdy w trójkącie wyznaczonym przez dwie ciemne ściany znajduje się jakaś inna ściana oraz gdy kąt między dwiema ciemnymi ścianami jest wklęsły.

Dwie ciemne ściany i inne ściany w środku

Załóżmy, że w środku trójkąta wyznaczonego przez dwie ciemne ściany AB i BC znajduje się jakaś inna ściana (a zatem też jakiś wierzchołek naszego wielokąta; patrz rysunek). Wybierzmy taki wierzchołek P spośród znajdujących się we wnętrzu ABC, żeby kąt BAP był jak najmniejszy, i poprowadźmy prostą AP do przecięcia z BC w punkcie Q. Teraz odcinki AB oraz BQ muszą być nieoświetlone oraz we wnętrzu trójkąta ABQ nie leży już żaden punkt wielokąta – a zatem, podobnie jak poprzednio, do trójkąta ABQ nie dociera światło, a w szczególności odcinek BP musi być nieoświetlony.



Zauważmy teraz, że odcinek BP rozcina cały wielokąt na dwie części. Skoro jest nieoświetlony, to i cała jedna część wielokąta (ta, w której nie będzie lampy) musi być nieoświetlona, bo każda droga od lampy do drugiej części poprowadzona wewnątrz wielokąta musiałaby przeciąć BP. Zatem albo wszystkie ściany od B do P po stronie zawierającej A, albo też po stronie zawierającej C, muszą być nieoświetlone; a stąd całą jedną część wielokąta będziemy mogli zastąpić ścianą BP.

Zauważmy, że powyższa operacja zastąpienia zmniejsza liczbę ścian w wielokącie co najmniej o jedną, a zatem operacji takich wykonamy co najwyżej n. Wobec tego możemy każdą z nich wykonywać w czasie $O(n)$ i nadal mieścić się w czasie. Zatem sprawdzenie, czy któryś z wierzchołków wielokąta leży we wnętrzu trójkąta ABC, wybranie punktu P, i wreszcie sprawdzenie, którą połowę wielokąta możemy usunąć, możemy zrealizować, po prostu przeglądając wszystkie możliwości.

Dwie ciemne ściany i kąt wklęsły

By osiągnąć sytuację, w której każda ciemna ściana sąsiaduje wyłącznie ze ścianami jasnymi, musimy jeszcze rozważyć przypadek, w którym dwie sąsiadujące ciemne ściany tworzą kąt wklęsły. Zakładamy przy tym, że pomiędzy żadnymi dwiema ciemnymi ścianami nie ma już kąta wypukłego, a zatem mamy pewien ciąg ścian ciemnych z kątami wklęsłymi pomiędzy każdą parą sąsiadów, a następnie po obydwu stronach tego ciągu ściany jasne. Pokażemy, że takiej sytuacji nie da się osiągnąć żadną pozycją lampy.

Niech A i B będą punktami styku ciągu ciemnych ścian ze ścianami jasnymi. Punkty tuż przy A i B są oświetlone, a zatem promienie świetlne poprowadzone z lampy do A i B idą we wnętrzu lub wzdłuż brzegów wielokąta. Powiedzmy, że lampa stoi w jakimś punkcie L. Punkt L nie może leżeć w jednej linii z A oraz B, bo wtedy – jako że pomiędzy A i B ściany ciemne tworzą kąt wklęsły – promień światła wychodziłby na zewnątrz wielokąta. Wiemy też, że skoro AL i BL nie przecinają boków wielokąta (mogą iść wzdłuż nich), to promień światła wypuszczony z L w jakimś kierunku pomiędzy AL i BL musi trafić w jakąś ścianę pomiędzy A i B. Ale ta ściana miała być nieoświetlona!

Czytelnikowi, który nie jest całkiem pewien, czy rozumie wszystkie detale, polecamy jako ćwiczenie zrozumieć, czemu powyższy argument nie działa, jeśli kąt pomiędzy ciemnymi ścianami jest wypukły.

Pojedyncze ciemne ściany

Doszliśmy zatem do sytuacji, w której każda ciemna ściana sąsiaduje po obu stronach ze ścianami jasnymi. Spróbujmy teraz powtórzyć powyższe rozumowanie – niech AB będzie ciemną ścianą, stawiamy lampę w jakimś punkcie L, światło z L dochodzi zarówno dowolnie blisko A, jak i dowolnie blisko B, czyli AL i BL nie opuszczają wielokąta (mogą iść wzdłuż jego brzegów). Gdyby A, B i L nie były współliniowe, to światło wypuszczone z L w jakimś kierunku pomiędzy AL i BL musiałoby zatrzymać się na ścianie AB (nie ma po drodze żadnej innej ściany wielokąta, bo żadna ściana nie przecina żadnego z trzech boków trójkąta ABL) – czyli przynajmniej kawałek AB byłby oświetlony. Zatem każda ciemna ściana wyznacza nam prostą, na której musi leżeć lampa.

To oznacza, że osiągnęliśmy sukces – po naszych redukcjach zarówno ściany jasne, jak i ściany ciemne wyznaczają łatwo wyrażalny warunek na położenie lampy (w przypadku ściany jasnej, lampa musi leżeć w półpłaszczyźnie wewnętrznej dla tej ściany, w przypadku ściany ciemnej – na prostej zawierającej tę ścianę). W szczególności, zauważmy, że oznacza to, że wielokąt otrzymany po redukcji musi już być gwiaździsty – a zatem spełnienie powyższych warunków będzie też wystarczające, by wszystkie jasne ściany były oświetlone.

Jeśli ścian ciemnych jest po redukcji więcej niż jedna, to przecinamy proste wyznaczone przez dowolne dwie z nich (są to różne proste, bo żadne trzy wierzchołki nie są współliniowe) i otrzymujemy dokładnie jednego kandydata na pozycję lampy, którego poprawność łatwo sprawdzić. W przypadku, gdy jest tylko jedna ściana ciemna, możemy łatwo sprawdzić, czy na prostej zawierającej tę ścianę możemy postawić lampę podobnie, jak robiliśmy to w przypadku samych ścian jasnych – patrzymy na półproste, które są przecięciami naszej prostej z półpłaszczyznami wyznaczonymi przez wszystkie pozostałe ściany (trzeba też pamiętać, że lampa nie może leżeć *na* ścianie, a zatem na koniec musimy sprawdzić, czy otrzymany odcinek możliwych pozycji lampy nie jest zawarty w ciemnej ścianie). Natomiast przypadek, gdy ścian ciemnych w ogóle nie ma, rozpatrzyliśmy już wcześniej.

Ostatecznie otrzymaliśmy rozwiązanie wzorcowe o złożoności czasowej $O(n^2)$. Można je znaleźć w plikach `gob.cpp`, `gob1.cpp` i `gob2.pas`.

Szybsze rozwiązanie

Czytelnikowi, który ma ochotę na odrobinę większe wyzwanie, można zaproponować próbę rozwiązania tego zadania w czasie $O(n \log n)$. Jak już wspomnieliśmy, sprawdzenie, czy wielokąt jest gwiaździsty, można przeprowadzić algorytmem przecięcia półpłaszczyzn. Pozostaje zatem problem redukcji ciemnych ścian, a konkretnie dwa problemy – sprawdzenie, czy we wnętrzu trójkąta znajduje się jakiś wierzchołek wielokąta, oraz sprawdzenie, który fragment wielokąta można usunąć. Istnieje kilka sposobów rozwiązania obydwu tych problemów, przedstawimy tu szkic jednego z nich.

Zauważmy, że ostatecznie naszym celem jest zawsze zredukowanie ciągu ścian ciemnych do pojedynczej ściany. Niech zatem A i B będą krańcami takiego ciągu. Jeśli ciąg ścian ciemnych „wystaje" poza odcinek AB (czego szczególnym przypadkiem jest ciąg ścian o kątach wklęsłych), to podobne rozumowanie, co w przypadku ciągu kątów wklęsłych, pokaże, że lampy ustawić się nie da. W przeciwnym razie, rozumowanie podobne do przypadku pojedynczej ciemnej ściany doprowadzi nas do wniosku, że lampa musi leżeć gdzieś na prostej AB, poza odcinkiem AB. Jeśli istnieje więcej niż jeden ciąg ścian ciemnych, to otrzymujemy dwie takie proste, i tylko jedno możliwe położenie lampy – a zatem będziemy mogli ograniczyć się do sprawdzenia poprawności takiego położenia (szczegóły takiego sprawdzenia w przypadku wielokąta niekoniecznie gwiaździstego pozostawimy niedopowiedziane). W przypadku, w którym ciąg do uproszczenia jest tylko jeden, możemy liniowo wyszukać punkt P jak powyżej (stać nas na to, gdyż zrobimy to tylko raz). Skoro punkt P jest krańcem ściany spoza naszego ciągu, to jest on oświetlony, tak więc zastępując jedną z części wielokąta przez ścianę AP lub BP, znajdziemy się w znanym już przypadku pojedynczej ciemnej ściany.

Testy

Przygotowano 10 plików testowych, każdy zawierający od kilku do kilkunastu testów. Małe testy (od *1* do *4*, w których $n \leqslant 20$) zostały wygenerowane ręcznie. Duże testy (od *5* do *10*) zostały wygenerowane dwuetapowo. Najpierw automatycznie wygenerowano wielokąty gwiaździste, a potem „popsuto” je ręcznie przy pomocy edytora graficznego. Ostatni test (*10*) zawiera wyłącznie ściany oświetlone.