Andrzej Walat
Treść zadania

Redakcja
Opracowanie

Jakub Łącki
Program

Dostępna pamięć: 64 MB. *OI, Etap II, dzień próbny, 10.02.2009*

Wyspy na trójkątnej sieci

Sieć trójkątów jest zbudowana z równobocznych trójkątów o boku 1 (ilustracja na końcu treści zadania). **Ścieżką** *na sieci trójkątów nazywamy dowolny skończony ciąg trójkątów (o boku 1) sieci, taki, że każde kolejne dwa trójkąty w tym ciągu mają wspólny bok.*

Figurę utworzoną przez punkty skończonej liczby trójkątów sieci nazywamy **wyspą***, jeśli dowolne dwa zawarte w niej trójkąty sieci można połączyć ścieżką utworzoną z trójkątów zawartych w tej figurze.*

Figury przedstawione na rysunkach 1.1, 1.2 i 1.3 są wyspami. Figura na rysunku 1.4 nie jest wyspą. Figury na rysunkach 2.2, 2.3 i 2.5 są przystające.

Zamierzamy dla każdego $n \leqslant 10$ opisać w systematyczny sposób wszystkie nieprzystające wyspy, jakie można utworzyć z n trójkątów o boku 1, i policzyć, ile ich jest.

Brzeg każdej wyspy, zbudowanej z co najwyżej dziesięciu trójkątów, jest łamaną zamkniętą złożoną z jednostkowych odcinków siatki i można go obiec (obrysować bez odrywania ołówka od papieru) w ten sposób, że dokładnie raz przebiegamy każdy odcinek brzegu i wracamy do punktu wyjścia. Może się zdarzyć, że trzeba będzie przy tym przejechać więcej niż raz przez ten sam punkt (patrz rysunek 2.4).

W przypadku wysp zbudowanych z co najwyżej dziesięciu trójkątów nie jest możliwa sytuacja taka, jak na rysunku 1.2, że brzeg figury składa się z dwóch rozłącznych części i nie można go obiec (obrysować bez odrywania ołówka od papieru).

Obiegając brzeg, po każdym jednostkowym odcinku wykonujemy skręt jednego z następujących typów:

- `a` — *o 120 stopni w lewo,*
- `b` — *o 60 stopni w lewo,*
- `c` — *o 0 stopni (tzn. brak skrętu),*
- `d` — *o 60 stopni w prawo,*
- `e` — *o 120 stopni w prawo.*

Każdy obieg brzegu wyspy można opisać za pomocą słowa złożonego z liter ze zbioru {`a`,`b`,`c`,`d`,`e`}*, odnotowując za pomocą odpowiedniej litery skręt, jaki należy wykonać po każdym kolejnym, jednostkowym odcinku łamanej tworzącej brzeg. Opis obiegu brzegu wyspy ma tyle liter, z ilu jednostkowych odcinków składa się łamana tworząca ten brzeg. Odnotowujemy skręt także po ostatnim odcinku łamanej, chociaż nie jest to konieczne dla jednoznacznego określenia jej kształtu. Ta, w pewnym sensie nadmiarowa, litera ułatwia przekształcenie danego opisu w opis innego obiegu tej samej figury, który zaczyna się w innym punkcie.*

Słowa `cdddcddd`, `dcdddcdd`, `cbbbcbbb` *opisują różne obiegi figury na rysunku 2.1.*

Słowa `cbeddcde`, `adcabcbb`, `abcbbadc` *opisują różne obiegi figury na rysunku 2.2.*

Słowa `acdabbcb` *i* `cddebced` *opisują różne obiegi figury na rysunku 2.3.*

Jeśli obiegając brzeg wyspy mamy stale jej wnętrze po prawej stronie, to mówimy, że jest to **obieg prawoskrętny***.*

Dla każdej wyspy można wyznaczyć zbiór wszystkich wysp do niej przystających oraz ich obiegi prawoskrętne. **Kodem wyspy** *nazwiemy słowo, które:*

1. *jest opisem pewnego prawoskrętnego obiegu brzegu jakiejś wyspy do niej przystającej,*
2. *jest wcześniejsze w porządku alfabetycznym od wszystkich innych słów spełniających warunek 1.*

Dla wysp przedstawionych na rysunkach 2.2 i 2.3, które są przystające, bierzemy pod uwagę wszystkie prawoskrętne obiegi obydwu:

`beddcdec`, `eddcdecb`, `ddcdecbe`, `dcdecbed`, `cdecbedd`, `decbeddc`, `ecbeddcd`, `cbeddcde`

oraz

`bcedcdde`, `cedcddeb`, `edcddebc`, `dcddebce`, `cddebced`, `ddebcedc`, `debcedcd`, `ebcedcdd`

i jako kod każdej z nich bierzemy to słowo, które w porządku alfabetycznym należy umieścić na pierwszym miejscu: `bcedcdde`.

Kodem wyspy przedstawionej na rysunku 2.4 jest słowo: `aadecddcddde`.

Napisz program, który:

- *dla podanego kodu wyspy o rozmiarze k wygeneruje kody wszystkich wysp o rozmiarze $k+1$, które można utworzyć przez dodanie do niej jednego trójkąta,*
- *dla podanej liczby całkowitej n wygeneruje kody wszystkich wysp rozmiaru n.*

Wejście

W pierwszym wierszu standardowego wejścia podana jest liczba całkowita t ($1 \leqslant t \leqslant 5$), oznaczająca liczbę zapytań. Każdy z kolejnych t wierszy zawiera opis zapytania pewnego typu. Zapytanie pierwszego typu składa się z litery `K` *oraz kodu wyspy składającej się z co najwyżej dziewięciu trójkątów, oddzielonych pojedynczym odstępem. Zapytanie drugiego typu składa się z litery* `N` *oraz liczby całkowitej n ($1 \leqslant n \leqslant 10$), oddzielonych pojedynczym odstępem.*

Wyjście

Na standardowe wyjście wypisz odpowiedzi na poszczególne zapytania.

Dla zapytań pierwszego typu należy najpierw wypisać liczbę różnych kodów wysp, które można utworzyć z przystających wysp opisanych podanym kodem poprzez dodanie jednego trójkąta. W następnym wierszu należy wypisać wszystkie te kody w kolejności alfabetycznej, pooddzielane pojedynczymi odstępami.

Dla zapytań drugiego typu należy wypisać liczbę różnych kodów wysp utworzonych z n trójkątów. W kolejnym wierszu należy wypisać wszystkie te kody w kolejności alfabetycznej.

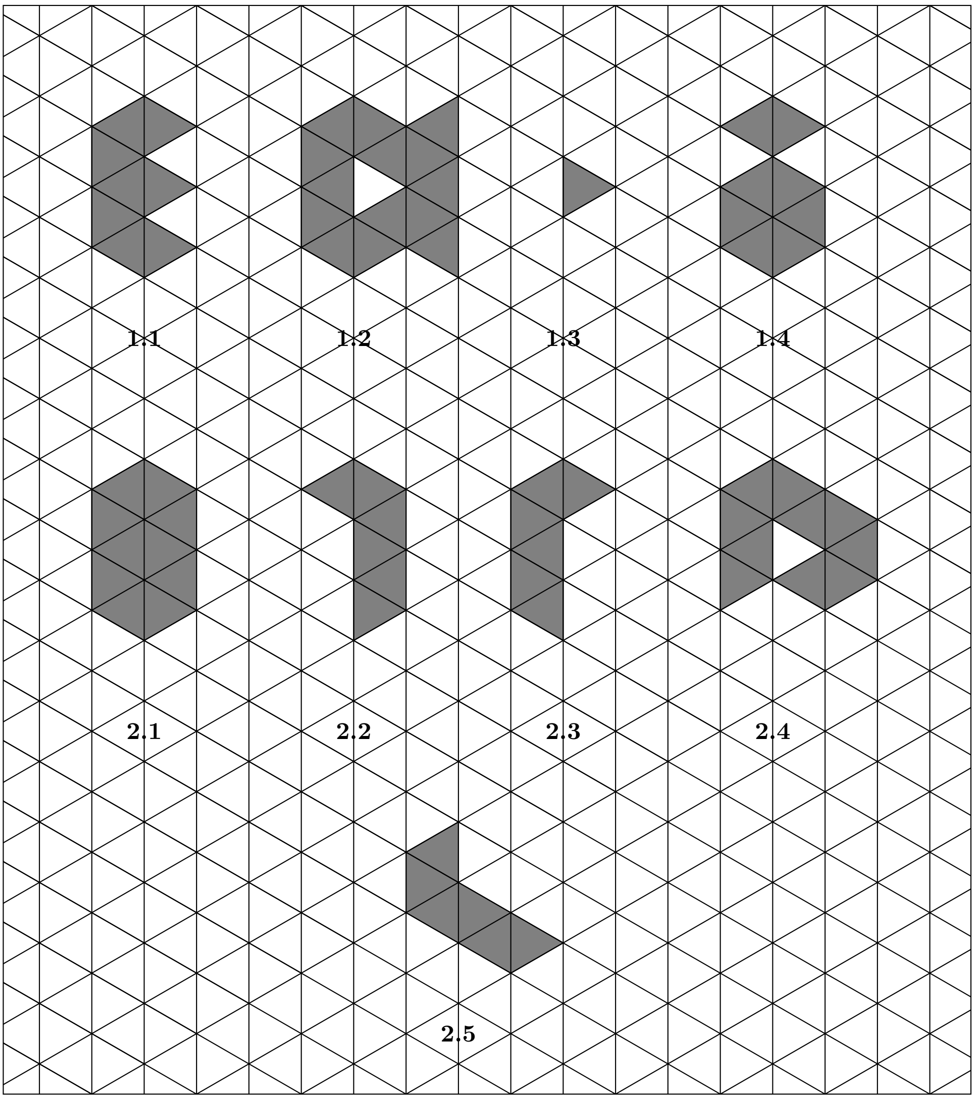
Przykład

Dla danych wejściowych:

```
2
K adeccecced
N 5
```

poprawnym wynikiem jest:

```
5
acedccecced addebcecced adebebecced adecbedcced cceccecce
4
aedddde bdecdde bececde ccedcde
```



Rozwiązanie

Zadanie o *Wyspach na trójkątnej sieci* zajmuje wyjątkowe miejsce wśród zadań olimpijskich, gdyż pojawiło się na zawodach Olimpiady Informatycznej już dwukrotnie: na II etapie pierwszej OI i teraz, na II etapie XVI OI. Celem ponownego wykorzystania tego zadania było m.in. porównanie, jak zmienił się charakter i poziom Olimpiady w ciągu tych piętnastu lat. Jakąś wskazówką w tym zakresie mogą być statystyki tego zadania z I Olimpiady — wówczas spośród 64 uczestników II etapu, troje uzyskało maksymalną liczbę 100 punktów, dziewięcioro miało ponad 90 punktów, a 0 punktów otrzymało 16 uczestników. Trudno jednak wysnuć jakieś konkretne wnioski z zestawienia tego typu statystyk, gdyż podczas owej olimpiady sposób sprawdzania był inny niż obecnie (m.in. przydzielano punkty uznaniowe, tudzież sprawdzano zachowanie programów na testach niezgodnych z opisem wejścia), no a poza tym zadanie o Wyspach nie było wtedy zadaniem próbnym. Dlatego dalsze dywagacje na ten temat pozostawiamy Czytelnikom.

Jako że opis rozwiązania tego zadania można znaleźć w książeczce I Olimpiady [1], nie publikujemy go w niniejszej książeczce. Ponieważ jednak publikacja [1] jest, niestety, w obecnych czasach trudno dostępna, więc umieszczamy dwie wskazówki dotyczące rozwiązania:

- Łączna liczba wszystkich wysp złożonych z co najwyżej 10 trójkątów jest stosunkowo mała (np. dla $n = 10$ mamy 448 takich wysp). Z tego względu odpowiadanie na zapytania drugiego typu można zrealizować za pomocą zapytań pierwszego typu.

- Dokładany trójkąt może mieć z wyspą 1, 2 lub 3 boki wspólne. Ostatnia z tych sytuacji ma jednak miejsce tylko dla jednego typu wyspy (2.4 na rysunku w treści zadania).

Poza powyższymi, trudność zadania jest czysto implementacyjna — w oryginalnym rozwiązaniu z I OI czyniono pewne sprytne spostrzeżenia dotyczące własności wysp, lecz można sobie poradzić i bez nich, np. konwertując wyspy z postaci tekstowej na wielokątową i nachalnie dodając jeden trójkąt wszędzie, gdzie się da. Po szczegóły odsyłamy Czytelników do programów wzorcowych, które reprezentują rozmaite podejścia do rozwiązania: `wys.java`, `wys1.pas`, `wys2.cpp` i `wys3.cpp`.

Dodajmy na koniec, że zestaw testów użytych na tegorocznych zawodach był inny niż oryginalny. Ułożono 10 testów, z czego testy 2, 4, 6, 7 i 8 zawierają tylko zapytania pierwszego typu, testy 1, 3 i 5 — drugiego typu, a testy 9 i 10 — obydwu typów.