

Zadanie: KOL

Kolorowy ogród



XXXIV OI, etap I. Plik źródłowy kol.* Dostępna pamięć: 256 MB.

12.10–16.11.2026

Uwaga: W tym zadaniu poznasz wynik punktowy swoich zgłoszeń zaraz po tym, jak Twoje programy zostaną ocenione przez system.

Ze względu na specyfikę tego zadania, zakazane jest dzielenie się testami do tego zadania w sekcji *Forum*. W tym zadaniu dostarczamy Ci narzędzie, dzięki któremu będziesz w stanie zwizualizować swoje rozwiązanie; jest ono dostępne w dwóch wersjach, na stronie https://oi.edu.pl/1/34oi_kol oraz jako skrypt w Pythonie do pobrania w SIO2.

Ogród Bajtazara jest pokryty kwadratowymi kostkami ułożonymi w wielki kwadrat o wymiarach 2001×2001 . Każda z kostek może świecić na jeden z k dostępnych kolorów, które dla uproszczenia numerujemy od 1 do k . Położenie kostki można opisać parą współrzędnych (x, y) , gdzie $-1000 \leq x, y \leq 1000$; współrzędna x oznacza numer kolumny kostek, od najbardziej zachodniej do najbardziej wschodniej, a współrzędna y numer rzędu kostek, od najbardziej południowego do najbardziej północnego. Bajtazar urządził z przyjaciółmi piknik na kocu o wymiarach 3×3 kostki, rozłożonym w samym środku ogrodu, a więc pokrywającym kostki o współrzędnych od $(-1, -1)$ do $(1, 1)$. Po chwili zgłodnieli i postanowili zamówić pizzę z popularnej *Karczmy pod Bajtem*. Karczma dostarcza pizzę za pomocą robotów.

Jeden z takich robotów wylądował za chwilę z pizzą w ogrodzie Bajtazara, na pewnej kostce o **nieznanych** współrzędnych (x_l, y_l) , gdzie $-400 \leq x_l, y_l \leq 400$, po czym będzie musiał dojechać na koc piknikowy. Tu właśnie potrzebna jest Twoja pomoc, ponieważ roboty te niestety słabo radzą sobie z orientacją w terenie: robot rozpoznaje kierunki świata, ale nie pamięta poprzednich obserwacji ani ruchów. Po wylądowaniu są w stanie jedynie rozpoznać kolor kostki, na której aktualnie się znajdują, oraz kolory czterech kostek sąsiadujących z nią bokami.

Na szczęście kostki w ogrodzie Bajtazara są zdalnie sterowane: możesz ustalić kolor każdej kostki przed przylotem robota. Możesz też wysłać robotowi program sterujący. Robot ma fabrycznie wbudowaną instrukcję, zgodnie z którą znajdując się na polu sąsiadującym bokiem z kocem piknikowym wjeżdża na koc (i podaje pizzę). Na każdym innym polu aktywuje się Twój program, który wyłącznie na podstawie koloru kostki pod robotem i kolorów czterech kostek wokół niego ma zdecydować, na którą z czterech sąsiednich kostek robot ma się przenieść.

Wejście

W pierwszym i jedynym wierszu wejścia znajdują się dwie liczby całkowite, k oraz t (gdzie $2 \leq k \leq 10^9$, $0 \leq t \leq 1$) oznaczające odpowiednio liczbę dostępnych kolorów kostek oraz rodzaj ograniczenia na miejsce lądowania robota. Dla $t = 0$ miejsce lądowania (x_l, y_l) spełnia warunek $|x_l|, |y_l| \leq 400$, a także na pewno nie znajduje się na kocu. Dla $t = 1$ wiemy dodatkowo, że $y_l = 0$.

Wyjście

W pierwszych 2001 wierszach należy wypisać po 2001 liczb całkowitych oddzielonych spacjami. Oznaczając one kolory kolejnych kostek: rzędy należy wypisywać w kolejności z północy na południe, a kostki w każdym rzędzie w kolejności z zachodu na wschód.

W następnym wierszu powinna znaleźć się liczba instrukcji dla robota q (gdzie $1 \leq q \leq 5 \cdot 10^6$).

W kolejnych q wierszach powinny znaleźć się instrukcje dla robota, po jednej w wierszu. Każda z tych instrukcji powinna składać się z pięciu liczb $k_{mój}$, k_N , k_E , k_S , k_W (każda z nich oznacza pewien kolor) oraz z jednej litery spośród N, E, S, W. Instrukcja taka mówi robotowi, że jeżeli kostka, na której robot stoi, oraz kostki sąsiadujące z nią od północy, wschodu, południa i zachodu mają odpowiednio kolory $k_{mój}$, k_N , k_E , k_S , k_W , to robot powinien przenieść się o jedną kostkę w kierunku opisanym przez podaną literę: północnym dla N, wschodnim dla E, południowym dla S, zachodnim dla W. Dla każdej piątki kolorów możesz wypisać co najwyżej jedną instrukcję.

Przykład

Dla danych wejściowych:

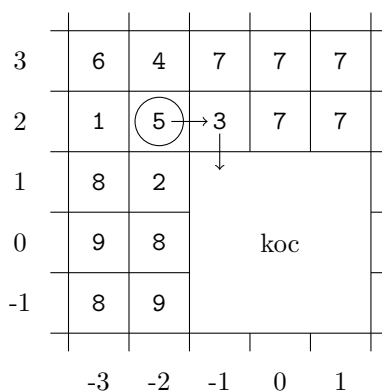
400 0

schemat rozwiązania to:

```
1 2 3 4 5 1 2 3 4 5 ... (2001 liczb)
... (kolejne 1999 wierszy)
6 7 8 9 10 6 7 8 9 10 ... (2001 liczb)
20
1 2 3 4 5 W
5 4 3 2 1 E
... (jeszcze 18 wierszy)
```

Dla powyższego schematu rozwiązania, udostępnionego w pliku `schemat.out`, robot trafi do Bajtazara z kostki o współrzędnych $(-2, 2)$, ale nie z każdej rozważanej kostki. W związku z tym program wypisujący zawartość tego pliku zostałby oceniony na 0 punktów.

Wyjaśnienie przykładu: W sytuacji przedstawionej na poniższym rysunku, robot (oznaczony kółkiem) znajduje się na kostce o współrzędnych $(-2, 2)$. Widzi pod sobą kolor 5, a na sąsiednich kostkach kolejno kolory 4, 3, 2, 1. Ma zatem zastosowanie druga instrukcja wypisana w powyższym przykładzie; robot przesunie się o jedną kostkę na wschód. Kostka ta sąsiaduje już z kocem, a więc w kolejnym kroku robot wjedzie na koc i poda Bajtazarowi pizzę (niezależnie od kolorów kostek znajdujących się pod kocem).



Wizualizator

W tym zadaniu udostępniliśmy Ci narzędzie pozwalające na wizualizację rozwiązania, a także częściowe sprawdzenie poprawności formatu wyjścia. Narzędzie jest dostępne na stronie https://oi.edu.pl/1/34oi_kol oraz jako skrypt `visualize.py`. Wyjście programu rozwiązującego zadanie należy zapisać do pliku, a następnie wysłać go przez stronę lub uruchomić polecenie

```
python visualize.py sciezka_do_pliku_wyjsciowego
```

W skrypcie `visualize.py` miejsce lądowania robota oraz tempo wyświetlania kolejnych kroków można ustawić modyfikując wartości zmiennych `START` oraz `DELAY` w kodzie źródłowym narzędzia.

Ocenianie

Zestaw testów dzieli się na następujące podzadania. Każde podzadanie składa się z jednego testu.

Podzadanie	Ograniczenia	Punkty
1	$k = 10^9, t = 0$	7
2	$k = 400, t = 0$	9
3	$k = 4, t = 0$	16
4	$k = 3, t = 0$	18
5	$k = 2, t = 1$	16
6	$k = 2, t = 0$	34

Punkty za dany test możesz otrzymać tylko wtedy, gdy z każdego możliwego miejsca lądowania (x_l, y_l) spełniającego warunki opisane w sekcji *Wejście*, robot wykonując instrukcje wypisanego przez Ciebie programu

dotrze na koc piknikowy – w szczególności nie zapętli się ani nie dojedzie do miejsca, w którym żadna z instrukcji nie będzie mogła być wykonana, w tym do miejsca na brzegu ogrodu. Jeśli to jest spełnione, liczbą punktów otrzymanych za test będzie minimum po miejscach lądowania z wartości $\lfloor p \cdot \frac{|x_l| + |y_l| + 16}{d} \rfloor$, gdzie p jest maksymalną liczbą punktów za test, natomiast d liczbą kroków, które wykonał robot dostarczając pizzę z danego miejsca lądowania (wliczając ostatni krok, w którym robot wjeżdża na koc). W szczególności maksimum punktów otrzymasz, jeśli z każdego rozważanego miejsca lądowania (x_l, y_l) robot wykonując Twój program dotrze do celu wykonując nie więcej niż $|x_l| + |y_l| + 16$ kroków (a jeśli wykona c razy więcej kroków, to otrzyma w przybliżeniu c razy mniej punktów).