Wojciech Rytter
Treść zadania, Opracowanie

Bartosz Kostka
Program

Dostępna pamięć: 128 MB. *OI, etap I, 6.10–3.11.2014*

Czarnoksiężnicy okrągłego stołu

Czarnoksiężnicy okrągłego stołu kolejny raz spotykają się na tajnej naradzie i kolejny raz nie mogą się zgodzić ze sobą, w jakiej kolejności powinni usiąść przy stole. W naradzie uczestniczy n czarnoksiężników. Każdy z nich jest jednoznacznie identyfikowany przez wysokość swojego spiczastego kapelusza. Wysokości kapeluszy są różnymi liczbami całkowitymi z zakresu od 1 do n (im wyższy kapelusz, tym większy staż czarnoksiężnika). Żeby nie zakłócić estetyki przy stole, wysokości kapeluszy czarnoksiężników siedzących obok siebie nie mogą się różnić o więcej niż p.

Ponadto nie wszyscy czarnoksiężnicy przepadają za sobą – jeśli czarnoksiężnik a nie lubi czarnoksiężnika b, to czarnoksiężnik b nie może siedzieć bezpośrednio po prawej stronie czarnoksiężnika a. Zakładamy, że przewodniczący narady (mający kapelusz o wysokości n) wybrał już swoje miejsce przy okrągłym stole. Na ile sposobów pozostali czarnoksiężnicy mogą do niego dołączyć?

Wejście

Pierwszy wiersz standardowego wejścia zawiera trzy liczby całkowite n, k i p ($1 \leqslant n \leqslant 1\,000\,000$, $0 \leqslant k \leqslant 100\,000$, $0 \leqslant p \leqslant 3$) pooddzielane pojedynczymi odstępami, oznaczające liczbę czarnoksiężników, liczbę informacji o ich niechęciach względem innych oraz maksymalną różnicę wysokości kapeluszy.

Kolejne k wierszy zawiera uporządkowane pary: i-ty z tych wierszy zawiera dwie liczby całkowite a_i i b_i ($1 \leqslant a_i, b_i \leqslant n$, $a_i \neq b_i$) oddzielone pojedynczym odstępem, oznaczające, że czarnoksiężnik w kapeluszu wysokości a_i nie lubi czarnoksiężnika w kapeluszu wysokości b_i. Każda uporządkowana para czarnoksiężników może się pojawić na wejściu co najwyżej raz.

W testach wartych 16% punktów zachodzi $n \leqslant 5$. W innych testach wartych 16% punktów zachodzi $p \leqslant 2$.

Wyjście

Jedyny wiersz standardowego wyjścia powinien zawierać liczbę całkowitą, będącą resztą z dzielenia przez $10^9 + 7$ liczby możliwości usadzenia czarnoksiężników.

Przykład

Dla danych wejściowych:

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

poprawnym wynikiem jest:

```
6
```

Wyjaśnienie do przykładu: *Czarnoksiężnicy mogą usiąść przy okrągłym stole na jeden z sześciu sposobów: 53124, 53142, 52143, 53412, 52314, 53214.*

Testy „ocen”:

`1ocen`: *mały test, w którym jeden z czarnoksiężników nie lubi nikogo;*

`2ocen`: $n = 5$, $k = 0$, $p = 3$;

`3ocen`: $n = 1\,000\,000$, $k = 0$, $p = 2$.

Rozwiązanie

Zadanie sprowadza się do policzenia cykli Hamiltona w skierowanym grafie n-wierzchołkowym, w którym krawędź między każdą parą wierzchołków u, v spełnia $|u - v| \leqslant p$ oraz niektóre krawędzie są zabronione (opowiadają konfliktom czarnoksiężników). Zajmiemy się jedynie przypadkiem $p = 3$; przypadki $0 \leqslant p \leqslant 2$ są dosyć proste.

Dla wygody jako nazw wierzchołków będziemy używać liczb będących odległością od n (czyli $0 := n$, $1 := n - 1$, $\ldots$). Tak więc liczbie n (przewodniczącemu narady) odpowiada teraz wierzchołek 0. Przyjmijmy, że przewodniczący narady siedzi na miejscu 0.

Oznaczmy przez X zbiór uporządkowanych par (i, j), gdzie $i, j \in [0, n - 1]$, reprezentujących konflikty (inaczej mówiąc, w naszych permutacjach nie może wystąpić para ze zbioru X). Trzeba jasno powiedzieć, że są to pary uporządkowane; jeśli np. $(9, 5) \in X$ oraz $(5, 9) \notin X$, to 9 nie może siedzieć obok 5 bezpośrednio z lewej strony, ale może siedzieć z prawej.

Tak więc rozważamy permutacje $\pi = (\pi_0, \pi_1, \ldots, \pi_{n-1})$ zbioru $\{0, \ldots, n - 1\}$, w których zawsze $\pi_0 = 0$, $\pi_{n-1} \in \{1, 2, 3\}$, każde dwa kolejne elementy różnią się co najwyżej o 3 oraz żadna para ze zbioru X nie występuje jako dwa kolejne elementy w π (wliczając π_{n-1}, π_0). Oznaczmy liczbę tych permutacji przez $ILE(n, X)$. Naszym zadaniem jest wyznaczenie tej liczby.

Niech $ILE_k(n, X)$ będzie liczbą tych permutacji spełniających powyższe warunki, od których dodatkowo żądamy, żeby $\pi_{n-1} = k$, gdzie $k \in \{1, 2, 3\}$. Oczywiście końcowy wynik to

$$ILE(n, X) = ILE_1(n, X) + ILE_2(n, X) + ILE_3(n, X).$$

Permutacje typu $(0 \diamond \diamond \ldots \diamond 1)$ i obliczanie pomocniczych zmiennych x_i, y_i

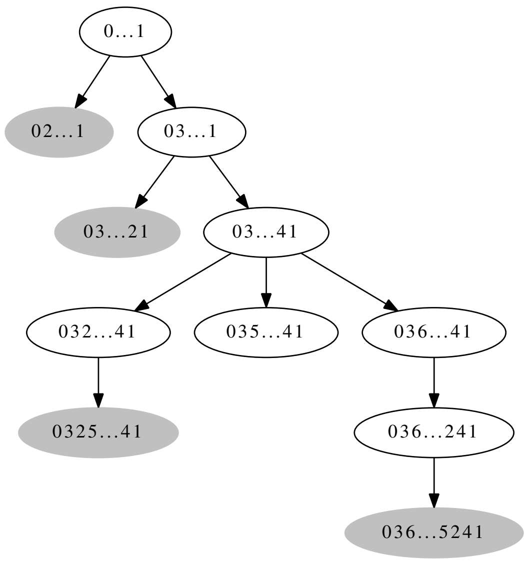
Definiujemy x_i, y_i jako liczby bezkonfliktowych permutacji liczb z przedziału $[i, n-1]$, w których pierwszy i ostatni wyraz należą do zbioru $\{i, i+1\}$. W przypadku x_i żądamy, aby pierwszym wyrazem było i, a w przypadku y_i, aby pierwszym wyrazem było $i+1$. Zauważmy, że $x_0 = ILE_1(n, X)$.

Obserwacja 1. Pozornie wydaje się, że zawsze $x_i = y_i$ (symetryczna definicja), ale nie zawsze tak jest, bo konflikty (zbiór X) nie muszą być symetryczne.

Naszym celem jest wyznaczyć wzór rekurencyjny na x_i oraz y_i. Zrobimy to na początek dla $i = 0$, przy założeniu, że $n \geqslant 8$. W tym celu przeanalizujemy możliwe prefiksy/sufiksy permutacji dające w sumie pewną liczbę kolejnych elementów (poczynając od 0). Symbol $\diamond$ oznacza *cokolwiek* (liczbę jeszcze nie wybraną). Jeśli pierwszym elementem jest 0 i ostatnim jest 1, to wszystkie możliwe sytuacje to:

$$0\,2 \diamond \ldots \diamond 1 \qquad 0\,3 \diamond \ldots \diamond 2\,1 \qquad 0\,3\,2\,5 \diamond \ldots \diamond 4\,1 \qquad 0\,3\,6 \diamond \ldots \diamond 5\,2\,4\,1.$$

Aby się o tym przekonać, możemy narysować drzewo, w którego każdym węźle na wszystkie możliwe sposoby próbujemy wydłużyć prefiks albo sufiks permutacji tak, by nie zaburzyć warunku, że każde dwa kolejne elementy różnią się co najwyżej o 3. Rozwijanie przerywamy, jeśli prefiks i sufiks zawierają w sumie k pierwszych elementów, jeden z nich kończy się liczbą $k-1$, a drugi $k-2$.



Jeśli pierwszym elementem jest 1 i ostatnim jest 0, to mamy sytuacje symetryczne:

$$1 \diamond \ldots \diamond 2\,0 \qquad 1\,2 \diamond \ldots \diamond 3\,0 \qquad 1\,4 \diamond \ldots \diamond 5\,2\,3\,0 \qquad 1\,4\,2\,5 \diamond \ldots \diamond 6\,3\,0.$$

Przyjmijmy na razie $X = \emptyset$. Powyższe sytuacje odnoszą się do $[0, n-1]$. Przeskalujmy je teraz do $[i, n-1]$ (zastępujemy zero przez i). Ogólnie dla $n - i \geqslant 8$ otrzymujemy:

$$x_i \;=\; y_{i+1} + y_{i+2} + y_{i+4} + y_{i+5}, \quad y_i \;=\; x_{i+1} + x_{i+2} + x_{i+4} + x_{i+5}. \tag{1}$$

Jeśli $X \neq \emptyset$, to niektóre składniki znikną. Na przykład jeśli $i = 0$ oraz $X = \{(0,2), (2,5)\}$, to

$$x_i \;=\; y_{i+2} + y_{i+4}, \qquad y_i \;=\; x_{i+1} + x_{i+2} + x_{i+4}.$$

Podsumowując, mamy:

$ILE_1(n, X) \;=\; x_0 \;=\; y_1 + y_2 + y_4 + y_5$ (bez składników konfliktowych ze zbioru X).

Permutacje typu $(0 \diamond \diamond \ldots \diamond 2)$ i $(0 \diamond \diamond \ldots \diamond 3)$

Przeanalizujmy teraz permutacje kończące się na 2. Podobnie jak poprzednio, możemy rozrysować drzewo możliwych konfiguracji typu prefiks/sufiks, które szybko się kończy. Ostatecznie dowiadujemy się, że permutacje takie mają dla $n \geqslant 8$ jedną z postaci:

$$0\,1 \diamond \ldots \diamond 2 \qquad 0\,3 \diamond \ldots \diamond 4\,1\,2 \qquad 0\,3\,1\,4 \diamond \ldots \diamond 5\,2.$$

Tak więc

$$ILE_2(n, X) \;=\; x_1 + x_3 + x_4 \quad \text{(ewentualnie bez składników konfliktowych).}$$

Podobnie rozumujemy dla permutacji kończących się na 3. W tym przypadku drzewo możliwości jest nieco większe. Ostatecznie okazuje się, że możliwe są tylko następujące konfiguracje:

$$0\,1\,2 \diamond \ldots \diamond 3 \quad 0\,1\,4 \diamond \ldots \diamond 5\,2\,3 \quad 0\,1\,4\,2\,5 \diamond \ldots \diamond 6\,3 \quad 0\,2\,1\,4 \diamond \ldots \diamond 3 \quad 0\,2\,5 \diamond \ldots \diamond 4\,1\,3.$$

Tak więc

$$ILE_3(n, X) \;=\; x_2 + x_4 + x_5 + y_3 + y_4 \quad \text{(ewentualnie bez składników konfliktowych).}$$

Rozwiązanie wzorcowe

Niech $\langle\alpha\rangle$ będzie funkcją zero-jedynkową, która sprawdza, czy permutacja α zawiera zabronioną parę ze zbioru X: jeśli zawiera, to wynikiem jest zero.

Poniższy pseudokod pokazuje, jak ostatecznie liczymy wynik końcowy:

Algorytm $ILE(n, X)$
  for $i := n$ **downto** $n-7$ **do** oblicz x_i, y_i brutalnie;
  for $i := n-8$ **downto** 0 **do begin**
    { oblicz x_i, korzystając z równania (1), dbając o to, by pominąć }
    { składniki wynikające z zabronionych par (ze zbioru X) }
    $x_i \;:=\; y_{i+1} \cdot \langle i+1, i, i+2\rangle + y_{i+2} \cdot \langle i+2, i+1, i, i+3\rangle +$
      $y_{i+4} \cdot \langle i+4, i+1, i, i+3, i+2, i+5\rangle + y_{i+5} \cdot \langle i+5, i+2, i+4, i+1, i, i+3, i+6\rangle$;
    analogicznie oblicz y_i;
  end

  { $wynik = ILE_1(n, X) + ILE_2(n, X) + ILE_3(n, X)$, gdzie $ILE_1(n, X) = x_0$ }
  $wynik \;:=\; x_0 + x_1 \cdot \langle 201\rangle + x_3 \cdot \langle 41203\rangle + x_4 \cdot \langle 520314\rangle + x_2 \cdot \langle 3012\rangle +$
    $x_4 \cdot \langle 523014\rangle + x_5 \cdot \langle 6301425\rangle + y_3 \cdot \langle 30214\rangle + y_4 \cdot \langle 413025\rangle$;
  return *wynik*;
end

Algorytm wykonuje liniową liczbę operacji arytmetycznych. Sprawdzanie konfliktów (implementacja funkcji $\langle\alpha\rangle$) działa także w czasie liniowym, gdyż każdy wierzchołek występuje w stałej liczbie (sensownych) par ze zbioru X.