Mittlerer Theorietag in Sarnen

Swiss Olympiad in Informatics

13. Februar 2019

Anleitung

Alle 4 Aufgaben werden auf die Gruppenmitglieder aufgeteilt. (Uniform zufällig.)

Bewertung

Die Präsentation sollte folgende Punkte enthalten:

1. Beschreibung der Idee und Erklärung des Algorithmus.
2. Beweis, wieso der Algorithmus korrekt ist.
3. Analyse der asymptotischen Laufzeit und des asymptotischen Speicherverbrauchs.

Die Bewertung beinhaltet hauptsächlich die Laufzeit und Korrektheit des Algorithmus. Die Qualität der Präsentation wird ebenfalls berücksichtigt.

Du darfst vorgeschriebenen Code zeigen, aber keine sonstigen Notizen. Zur Erklärung steht dir ein leeres Blatt Papier zur Verfügung.

Pro Aufgabe solltest du nicht länger als 10 Minuten benötigen.

Technische Hinweise

Es steht nicht, nach welchen Kriterien du das Programm optimieren musst. Es sollte aber in allen Fällen klar sein, welcher Parameter gemeint ist (z.B. Zahl n oder Länge des Eingabestrings).

Du kannst annehmen, dass arithmetische Operationen mit Ganzzahlen in konstanter Zeit berechnet werden können, unabhängig von ihrer Grösse. Die Darstellung einer Zahl benötigt $O(1)$ Speicher.

Fünf Nächte Werwolf

Maus Daniel hat ein neues Spiel erfunden: es heisst fünf Nächte Werwolf. Maus Bibin würde auch gerne mitspielen, aber da gerade der Game Jam läuft, bekommt er nicht sonderlich viel von den Regeln mit. Was er mitbekommt ist:

Am Anfang stehen alle n Mäuse in einer Reihe und haben alle eine verschiedene Geheimzahl von 11 bis n. Alle dürfen höchstens eine Nacht auswählen, bei der sie zu einem beliebigen Platz in der Reihe gehen (und nehmen ihre Geheimzahl mit). Am Ende jeder Nacht werden alle Geheimzahlen (nun in der angepassten Reihenfolge) aufgedeckt und alle Mäuse gehen anschliessend wieder an ihren Ursprungsplatz zurück.

Aufgabenstellung

Du kennst n, die Anzahl der Mäuse und die fünf Sequenzen der Geheimzahlen wie sie am Ende der Nächte waren $a_1, \ldots a_n, b_1, \ldots b_n, \ldots e_1, \ldots, e_n$. Maus Bibin weiss zwar nicht, was genau das Spielziel ist, jedoch will er gerne wissen, was eine mögliche Startaufstellung der Mäuse ist. Hilf ihm eine mögliche Startaufstellung herauszufinden. (Falls es mehrere mögliche Startaufstellungen gibt ist Bibin mit jeder davon zufrieden.)

Beispiel

$n = 3, 123; 123; 123; 123; 231$ (Antwort: 1 2 3; falls alle Mäuse entschieden haben sich in der letzten Nacht zu bewegen, könnte das die Startposition gewesen sein.)

Brücken

Gegensätzlich zum Glauben von vielen ist Sarnen ein kleiner Ort im Meer, der aus mehreren Inseln besteht. Sie haben herausgefunden, dass es schwierig ist, sich zwischen den Inseln zu bewegen. Daher hat der Bürgermeister Maus Benjamin den Plan, Brücken zu bauen, sodass diese dann alle Inseln miteinander verbinden. Er hat herausgefunden, zwischen welchen Inseln es überhaupt möglich ist, Brücken zu bauen und hat schon ein Angebot von Stofls Brückenbau Unternehmen für die einzelnen Brücken erhalten.

Maus Stefanie hat herausgefunden, dass Maus Benjamin die Brücken in Sugus bezahlen wird und daher will sie auch eine Brücke bauen. Ausserdem gilt

- Maus Benjamin wird die Brücken so auswählen, sodass die total Kosten minimiert werden.
- Niemand weiss, was in Benjamins Kopf vor sich geht: wenn es mehrere Möglichkeiten gibt, die Kosten zu minimieren, wird er eine beliebige davon auswählen.
- Maus Stefanie darf nur für eine Brücke ein Angebot machen.

Aufgabenstellung

Du bekommst, n, die Anzahl Inseln, m, die Anzahl Brücken die gebaut werden können und u_i, v_i, c_i Start und Endpunkt der Brücke resp. die Anzahl Sugus die Stofl für den Bau verlangt. Hilf Maus Stefanie herauszufinden, wie viele Sugus sie sicher bekommen kann, wenn sie ein Angebot für eine Brücke macht.

Beispiel

$n = 3, m = 3, 0, 1, 3; 0, 2, 5; 1, 2, 10$ (Antwort: 4; Maus Stefanie kann ein Angebot für die Brücke von 0 zu 2 machen für 4 Sugus.)

Aufsteigend

Es sei $D_{n,k}$ eine Liste aller *aufsteigenden* Folgen von n ganzen Zahlen mit Elementen zwischen 1 und k. Zum Beispiel:

$$D_{3,4} = (1,2,3),\ (1,2,4),\ (1,3,4),\ (2,3,4)$$
$$D_{2,5} = (1,2),\ (1,3),\ (1,4),\ (1,5),\ (2,3),\ (2,4),\ (2,5),\ (3,4),\ (3,5),\ (4,5)$$

In jeder List $D_{n,k}$ haben die einzelnen Folgen eine festgelegte Reihenfolge: die *lexikographische* Ordnung. Diese Ordnung wird bei beiden der obigen Beispiele verwendet.

Formal ist die lexikographische Ordnung wie folgt definiert:

Die Folge $a_0, \ldots, a_{n-1}$ ist *lexikographisch kleiner* als die Folge $b_0, \ldots, b_{n-1}$ falls es einen index $j \geq 0$ gibt sodass $a_j < b_j$ und wir für alle $i < j$ haben dass $a_i = b_i$.

Zum Beispiel, $(3, 7, 12)$ ist lexikographisch kleiner als $(3, 8, 9)$. Also würde in $D_{3,20}$ die Folge $(3, 7, 12)$ früher auftreten als $(3, 8, 9)$. Die Folge $(3, 7, 12)$ ist auch kleiner als $(4, 5, 6)$.

Task

Du bekommst n, k, und eine Folge S, welche eine der Folgen in $D_{n,k}$ ist. Deine Aufgabe ist es, den Index von S in $D_{n,k}$ zu finden. (Die erste Folge in $D_{n,k}$ hat den Index 1.)

Zum Beispiel, in $D_{3,2}$ ist der Index von $(1, 2, 4)$ gleich 2 und der Index von $(2, 3, 4)$ ist 4.

Beispiel

$n = 5, k = 8, S = [1, 2, 3, 4, 8]$ (Antwort: 4; die ersten vier Folgen in $D_{5,8}$ sind $(1, 2, 3, 4, 5)$, $(1, 2, 3, 4, 6)$, $(1, 2, 3, 4, 7)$, und **(1,2,3,4,8)**).

Zyklische Verschiebung

Die i-te *zyklische Verschiebung* der Sequenz $A = (a_0, \ldots, a_{n-1})$ ist die Sequenz, die entsteht, wenn man die Elemente von A um i Positionen nach links verschiebt und den Anfang mit dem Ende verbindet. Formell, für $0 \leq i < n$ ist die i-te zyklische Verschiebung definiert als $C(A, i) = (a_i, \ldots, a_{n-1}, a_0, a_1, \ldots, a_{i-1})$.

Zum Beispiel für $A = (3, 4, 5, 6)$ gibt es vier zyklische Verschiebungen:

$$\begin{aligned} C(A,0) &= (3,4,5,6) \\ C(A,1) &= (4,5,6,3) \\ C(A,2) &= (5,6,3,4) \\ C(A,3) &= (6,3,4,5) \end{aligned}$$

Beachte, dass für einige Sequenzen verschiedene zyklische Verschiebungen dem selben Resultat entsprechen können. Zum Beispiel, falls $B = (1, 2, 1, 2)$ dann $C(B, 0) = C(B, 2) = (1, 2, 1, 2)$ und $C(B, 1) = C(B, 3) = (2, 1, 2, 1)$.

Die zyklische Verschiebung einer Sequenz A kann *lexikographisch geordnet* werden. Das kleinste Element in dieser Ordnung nennen wir die *lexikographisch kleiste zyklische Verschiebung*.

Genauer: die lexikographische Ordnung ist definiert als:

Die Sequenz $X = (x_0, \ldots, x_{n-1})$ ist *lexikographisch kleiner* als die Sequenz $Y = (y_0, \ldots, y_{n-1})$ falls es einen Index $j \geq 0$ gibt, sodass $x_j < y_j$ und für alle $i < j$ gilt $x_i = y_i$. Wir schreiben dies als $X \prec Y$.

Im Besipiel oben gilt $C(A, 0) \prec C(A, 1) \prec C(A, 2) \prec C(A, 3)$ und $C(B, 0) = C(B, 2) \prec C(B, 1) = C(B, 3)$. Für die Sequenz $C = (3, 1, 3, 7)$ ist die lexikographische Ordnung ihrer zyklischen Verschiebungen $C(C, 1) \prec C(C, 0) \prec C(C, 2) \prec C(C, 3)$. Beachte, dass $C(C, 0) = (3, 1, 3, 7) \prec (3, 7, 3, 1) = C(C, 2)$.

Also ist die lexikographisch kleinste Verschiebung von A die Verschiebung um 0, für C die Verschiebung m 1 und für B gibt es mehrere Verschiebungen, welche der kleinsten lexikographischen Verschiebung entsprechen: 0 und 2.

Aufgabe

Du kennst n $(1 \leq n \leq 100\,000)$, die Länge einer versteckten Sequenz $A = (a_0, a_1, \ldots a_{n-1})$.

Deine Aufgabe ist es zu bestimmen welche zyklischen Verschiebungen die kleinsten sind. Allerdings kennst du die Sequenz nicht. Um die Informationen zu erhalten, die du benötigst, musst du Fragen zur Sequenz stellen. Für jede Frage wählst du zwei Indizes i und j $(0 \leq i, j < n)$ und rufst die Funktion CMP(i, j) auf. Diese gibt entweder <, = oder > zurück, je nach dem ob $a_i < a_j$, $a_i = a_j$ resp. $a_i > a_j$.

Finde heraus, für welches k die zyklischen Verschiebungen $C(A, k)$ lexicographisch minimal ist. Falls es mehr als eine Lösung gibt, kannst du eine beliebige auswählen.

Minimiere in erster Linie die asymptotische Anzahl Aufrufe von CMP, in zweiter Linie Laufzeit/Speicher.