

Catalysts Christmas Challenge

Linz / Austria

December 18, 2009

Siegerprämie / Prize

Catalysts

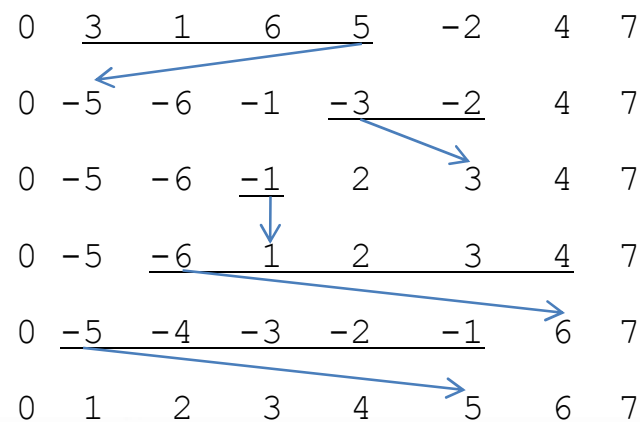
- Unsere Sponsoren finanzieren einen tollen Preis, den wir im Anschluss an den Wettbewerb verschenken.
- Um daran teilnehmen zu können, müssen Sie der Weitergabe Ihrer Kontaktdaten an die Sponsoren zustimmen.
- An der „Catalysts Coding Challenge“ können Sie natürlich vollkommen anonym teilnehmen.
- Es kann kein Bezug hergestellt werden zwischen dem Wettbewerb und dem Gewinnspiel.
- Our sponsors provide the funds for a great prize that we give away after the contest.
- In order to be eligible to get the prize you have to agree that we can forward your contact data to our sponsors.
- Your participation in the “Catalysts Coding Challenge” is completely anonymous.
- There is no way of referencing between the challenge and the lottery.

- Nach dem Wettbewerb!
- After the challenge!
- Wie beim Advent-Gewinnspiel:
 - Bei taskmind registrieren
 - Aufgabe „Am CCC Gewinnspiel“ an den Benutzer Gewinnspiel schicken
- Like with the Advent Lottery:
 - Register with taskmind
 - Send the task „Participate in the CCC Lottery“ to the user „Gewinnspiel“

<http://www.catalysts.cc/events/2009/advent-gewinnspiel/teilnahme/>

Molekulare Uhr / Molecular Watch Catalysts

- Es geht um die Berechnung der minimalen genetischen Distanz zwischen Virenstämmen.
- Biologen geben Genen Namen. Wir verwenden statt dessen numerische IDs. Statt *Fi Muok La GH ...* schreiben wir 3 1 6 5 ...
- Genabschnitte sind linksdrehend (negatives Vorzeichen) oder rechtsdrehend (positives Vorzeichen).
- **Gesucht ist die minimale Anzahl an Schritten um eine beliebige Permutation zu sortieren.**
- Ein Schritt ist dabei definiert durch das Vertauschen eines Genabschnitts mit gleichzeitigem Vorzeichenwechsel der IDs
- The puzzle is about computing the minimal genetic distance between viruses.
- Biologists give names to genes. We use numeric IDs instead. Instead of *Fi Muok La GH ...* we write 3 1 6 5 ...
- Gene sequence either is left turning (negative sign) or right turning (positive sign).
- **The goal is to find the minimal number of steps to sort an arbitrary permutation.**
- One step is defined as reversing a sub-sequence of the permutation including also the sign changes.



Molekulare Uhr / Molecular Watch

Catalysts

- Das Rätsel ist auf mehrere Levels aufgeteilt – zum Schluss können Sie die genetische Distanz berechnen.
 - Auf der vorigen Seite sehen Sie den Input für den letzten Level.
 - In den folgenden Levels
 - können Sie sich entweder auf die geforderte Angabe im jeweiligen Level beschränken (und somit schneller vorankommen), oder
 - können Sie schon versuchen, kompliziertere Anforderungen zu berücksichtigen, die Sie erkennen können bzw. erraten (und somit vorne weg mehr Arbeit haben und sich eventuell später was sparen).
 - Denken Sie daran, dass ein abgeschlossener Level 1 mehr wert ist, als ein nicht abgeschlossener Level 1, in dem halt schon Features aus späteren Levels berücksichtigt sind.
 - Der Gewinner ist, wer als erster den letzten Level geschafft hat bzw. wer am schnellsten am weitesten gekommen ist.
 - Die Aufgabe ist schwer – auch wenn man nur ein paar Levels schafft, kann man schon sehr stolz auf sich sein!
- The puzzle is split into separate levels – in the end you can calculate a genetic distance.
 - On the previous page you see the input for the last level.
 - When working through the levels
 - you can either concentrate on just the requirements of the respective level (hence proceed as quickly as possible), or
 - you can try to consider more complicated requirements that you can discern or foresee (hence do more work early on and potentially reap the benefits later).
 - Keep in mind that a finished level 1 is worth more than an unfinished level 1 where you tried to consider more complicated features from later levels.
 - The winner is who is the first to solve the last level or who is the fastest to get through the most levels.
 - The problem is hard – even if you only complete a couple of levels you can be very proud of yourself!

- Die Grundannahme der Genforscher für Mutationen lautet: Je 2 Arten besitzen immer einen gemeinsamen Vorläufer.
- 2 Arten einer Gattung besitzen denselben Satz von Genen, jedoch in unterschiedlicher Reihenfolge. Ursache sind gelegentliche „**Inversionen**“. Ein Teilabschnitt wird aus der DNA ausgeschnitten und in umgekehrter Reihenfolge wieder eingefügt.
- Diese genetische Veränderungen benutzen Forscher, um Artenstammbäume zu rekonstruieren.
- Eine höhere Anzahl von Inversionen deutet auf eine größere evolutionäre Distanz hin zwischen zwei Arten, so sind z.B. 3 Inversionen erforderlich, um Kohl in Rüben umzuwandeln,.
- Deine Aufgabe ist es einen **effizienten** Algorithmus zu schreiben, der die **minimale** Abfolge von Inversionen rechnet.
- Tipp: Mit bekannten Sortier-Algorithmen erreichen Sie das Ziel nicht!
- The root assumption of gene scientists for mutations is: 2 species always have a common ancestor.
- So 2 species of one genus have the same set of genes, but in a different order. The reason is occasional **inversions**. A DNA sequence is cut out and inserted again in reversed order.
- Scientists use this information to reconstruct family trees of a species.
- A larger number of inversions hints at a larger evolutionary distance between 2 species, e.g. 3 inversions are necessary to turn cabbage into beetroot.
- Your task is to write an **efficient** algorithm which calculates the **minimal** number of inversions necessary to get from one species to another species.
- Hint: You won't reach the goal with typical sorting algorithms!

Level 1

Finden Sie sogenannte „orientierte Paare“ in einer
Permutation $P = (x_0 x_1 x_2 \dots x_{n-1})$

- Ein orientiertes Paar
 - ist ein Paar von (nicht unbedingt unmittelbar) aufeinander folgenden Zahlen mit $|x_i| - |x_j| = \pm 1$
 - besteht immer aus einer positiven und einer negativen Zahl
- $P = (3 \ 1 \ 6 \ 5 \ -2 \ 4)$ hat 2 orientierte Paare: $(1, -2)$ und $(3, -2)$
- Die einzelnen Zahlen in der Eingabe sind jeweils durch ein Leerzeichen getrennt (gilt für alle Levels).
- Eingabe: PermutationsLänge Permutation
 - PermutationsLänge = Integer.
 - Permutation = {Integer}.
- Ausgabe: OPListe // aufsteigend nach x sortiert
 - OPListe = AnzahlPaare {Paar}.
 - AnzahlPaare = Integer
 - Paar = x y. // x und y sind Integers
 - Die OPListe ist nach x aufsteigend sortiert

Input: 6 3 1 6 5 -2 4

Find so-called “oriented pairs” in a permutation
 $P = (x_0 x_1 x_2 \dots x_{n-1})$

- An oriented pair
 - is a pair of consecutive integers (not necessarily directly after each other) with $|x_i| - |x_j| = \pm 1$
 - always consists of a negative number and a positive one
- Permutation $P = (3 \ 1 \ 6 \ 5 \ -2 \ 4)$ has 2 oriented pairs: $(1, -2)$ and $(3, -2)$
- The numbers in the input are separated with a space (also in later levels).
- Input: PermutationLength Permutation
 - PermutationLength = Integer.
 - Permutation = {Integer}.
- Output: OPList // sorted by x, ascending
 - OPList = NumberOfPairs {pairs}.
 - NumberOfPairs = Integer.
 - Pair = x y. // x and y are integers
 - The OPList is sorted by x in ascending order

Output: 2 1 -2 3 -2