



Heidi's Calling Codes

by Cheng Zhong

In the 31st century, more and more new countries emerge, and the available Country Calling Codes (CCCs in short) are fewer and fewer. The CCC specifies the country which the caller is dialing, such as “41” for Switzerland, “1” for U.S. and “218” for Libya at the beginning of the 21st century.

In order to ensure the uniqueness of a phone number in the world, CCCs must be prefix-free codes. This means, in a valid set of CCCs, no CCC could be the prefix of any other CCC. For example, if we use “41” for Switzerland, then we can't use “4” or “41” or “418” for China. Otherwise, when someone dials “41888888”, the system could not tell which the destination country is.

Heidi has just received the task to maintain the CCC system. Could you help her with some subtasks? Please note that in the 31st century, people use binary phone numbers instead of decimal phone numbers, because the phone is always dialed by computer. The CCCs are consequently binary codes. And the length of each CCC may be as large as 20.

Easy Task — Prefix

Given a set of CCCs, decide whether they are *valid*, i.e. whether they are prefix-free codes.

To do so, write a function `bool isValidPrefixSet(int n, char* prefix[])` that

- Takes as parameters
 - *n* : the number of CCCs
 - *prefix* : an array with *n* CCCs at positions 0 to *n* - 1
- Returns 1 (true) for *valid*, or 0 (false) for not.

Sample Input and Output

For the set {10, 0, 111}, you should return 1.



Heidi et les Indicateurs Téléphoniques

par Cheng Zhong

Au 31^e siècle, de plus en plus de pays émergent et les indicateurs téléphoniques internationaux (ITI pour faire court) disponibles sont de moins en moins nombreux. Un ITI spécifie le pays qu'on appelle, par exemple "41" pour la Suisse, "1" pour les Etats-Unis et "218" pour la Libye au début du 21^e siècle.

De sorte à assurer le caractère unique des numéros de téléphone dans le monde, les ITI doivent être des codes sans préfixes. Ceci veut dire que dans un ensemble valide de ITI, aucun ITI ne peut être le préfixe d'un autre ITI. Par exemple, si on utilise "41" pour la Suisse, alors on ne peut pas utiliser "4", "41" ou "418" pour la Chine. Autrement, lorsque quelqu'un compose le "41888888", le système serait incapable de savoir dans quel pays se trouve le destinataire de l'appel.

On vient de confier à Heidi la manutention de l'ensemble du système des ITI. Pouvez-vous l'aider avec certaines tâches ? Veuillez remarquer qu'au 31^e siècle, les numéros de téléphone binaires ont supplanté les numéros décimaux, car désormais le téléphone est toujours utilisé via un ordinateur. Les ITI sont par conséquent des codes binaires et la longueur de chaque ITI peut arriver jusqu'à 20 bits.

Tâche Facile — Indicateurs d'Appel

Étant donné un ensemble d'ITI, décidez s'il est *valide*, c'est à dire si tous les codes sont sans préfixes. Pour cela, écrivez une fonction `bool isValidPrefixSet(int n, char* prefix[])` qui

- Prend les paramètres suivants
 - *n* : le nombre de ITIs.
 - *prefix* : un vecteur avec *n* ITIs aux positions 0 à *n* - 1
- Retourne 1 (true) pour *valide*, ou 0 (false) sinon.

Exemple d'Entrées et Sorties

Pour l'ensemble {10, 0, 111}, vous devriez retourner 1.



Handys mit Chaotischen Codes

von Cheng Zhong

Im 31. Jahrhundert entstehen ständig neue Länder, was dazu führt, dass die freien Ländervorwahlen (CCC, für Country Calling Code) knapp werden. Eine Ländervorwahl legt fest, in welchem Land sich das angerufene Telefon befindet. Zum Beispiel haben wir „41“ für die Schweiz, „1“ für die Vereinigten Staaten, und „218“ für Lybien zu Beginn des 21. Jahrhunderts.

Damit eine Telefonnummer weltweit eindeutig ist, müssen CCCs Präfix-frei sein. Das bedeutet, dass in einer gültigen CCC-Menge kein Code ein Präfix eines anderen Codes sein kann. Verwenden wir beispielsweise „41“ für die Schweiz, so können wir nicht „4“, „41“ oder „418“ für China verwenden. Sonst könnte das System die Telefonnummer „41888888“ nicht eindeutig einem Land zuordnen.

Heidi hat den Auftrag erhalten, die Ländervorwahlen zu verwalten. Könntet ihr Heidi bei einigen Aufgaben helfen? Dazu müsst ihr wissen, dass im 31. Jahrhundert binäre Telefonnummern verwendet werden, weil Telefone nur noch maschinell bedient werden. Dadurch sind CCCs binäre Codes. Ein CCC ist bis zu 20 Ziffern lang.

Einfache Aufgabe — Präfix

Entscheidet, ob eine gegebene Menge CCCs *gültig* ist, d. h. ob die Codes Präfix-frei sind.

Schreibt dazu eine Funktion `bool isValidPrefixSet(int n, char* prefix[])` welche

- Folgende Parameter nimmt
 - *n* : Anzahl der CCCs
 - *prefix* : Ein Array mit *n* CCCs auf Positionen 0 bis *n* – 1
- 1 (true) für *gültig* zurückgibt, und sonst 0.

Beispieleingabe und -ausgabe

Für die Menge {10, 0, 111} soll eure Funktion 1 zurückgeben.