



Heidi's Calling Codes

Heidi et les Indicatifs Téléphoniques

Handys mit Chaotischen Codes

Cheng Zhong

Medium Task — Prefix

Given a *valid* set of CCCs, decide whether it is *full*. Here *full* means that we could not add any other candidate CCC into the set and keep the validity in the meantime.

To do so, write a function `bool isFullPrefixSet(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 *full*, or 0 (false) for not.

Sample Input and Output For the set {10, 0, 110, 111}, you should return 1.

Tâche Moyenne — Préfixe

Étant donné un ensemble d'ITI *valide*, décidez s'il est *complet*. Ici *complet* veut dire qu'on ne peut ajouter d'autres ITI à l'ensemble et faire en sorte qu'il reste valide en même temps.

Pour cela, écrivez une fonction `bool isFullPrefixSet(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 *complet*, ou 0 (false) sinon.

Exemple d'Entrées et Sorties Pour l'ensemble {10, 0, 110, 111}, vous devriez retourner 1.

Mittlere Aufgabe — Präfix

Entscheidet, ob eine gegebene, *gültige* Menge CCCs *voll* ist. Eine Menge ist *voll*, wenn kein weiterer Code hinzugefügt werden kann, ohne dass die Menge ungültig wird.

Schreibt dazu eine Funktion `bool isFullPrefixSet(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 *voll* zurückgibt, und sonst 0.

Beispieleingabe und -ausgabe Für die Menge {10, 0, 110, 111} sollte eure Funktion 1 zurückgeben.