Seletiva para Maratona de Programação de 2014

Comentários sobre os problemas

Patrocínio:

Departamento de Ciência da Computação IME-USP

Problema A:

Autor do problema: <autor problema>
Análise: <autor analise>

Problema B: Renzo e a decoração capicuânica

Autor do problema: Marcio T. I. Oshiro / Carlinhos
Análise: Marcio T. I. Oshiro

Este problema pode ser divido em duas partes: encontrar a representação de N em uma base b; e verificar se essa representação é capicua (palíndromo).

A representação de N em uma base $b > 0$ é $(a_m a_{m-1} \ldots a_1 a_0)_b$ tal que $a_i \in \{0, 1, \ldots, b-1\}$, para todo $i \in \{0, 1, \ldots, m\}$, $a_m \neq 0$ e

$$N = a_m b^m + a_{m-1} b^{m-1} + \cdots + a_1 b + a_0.$$

Note que $N = b(a_m b^{m-1} + a_{m-1} b^{m-2} + \cdots + a_1) + a_0$. Logo a_0 é o resto da divisão de N por b. De forma análoga, temos que a_1 é o resto da divisão de $(N - a_0)/b$ [1] por b. Repetindo esse raciocínio, podemos facilmente encontrar o valor de a_i, para todo $i \in \{0, 1, \ldots, m\}$.

Dado a sequência $(a_m, a_{m-1}, \ldots, a_1, a_0)$, falta apenas verificar se $a_{m-i} = a_i$, para $i \in \{0, 1, \ldots, \lfloor m/2 \rfloor\}$. Se todos os pares forem iguais, a representação de N na base b é capicua. Caso contrário, não é.

Como $N < 2^{31}$, vale que $m \leq 31$, pois quanto menor a base, mais dígitos são necessários para representar o mesmo número.

Problema C:

Autor do problema: <autor problema>
Análise: <autor analise>

Problema D:

Autor do problema: <autor problema>
Análise: <autor analise>

Problema E:

Autor do problema: <autor problema>
Análise: <autor analise>

Problema F:

Autor do problema: <autor problema>
Análise: <autor analise>

Problema G:

Autor do problema: <autor problema>
Análise: <autor analise>

Problema H:

Autor do problema: <autor problema>
Análise: <autor analise>

[1] Note que $(N - a_0)/b$ corresponde à divisão inteira de N por b.

Problema I:

Autor do problema: <autor problema>
Análise: <autor analise>

Problema J:

Autor do problema: <autor problema>
Análise: <autor analise>

Problema K:

Autor do problema: <autor problema>
Análise: <autor analise>

Problema L: Emplacando os *tuk-tuks*

Autor do problema: Marcio T. I. Oshiro / Carlinhos
Análise: Marcio T. I. Oshiro

Este problema é bem direto, basta contar a quantidade máxima de placas com C consoantes e D dígitos, lembrando que as consoantes sempre aparecem antes dos dígitos.

Existem 26 escolhas de consoantes e 10 escolhas de dígitos. Logo, a resposta é

$$26^C \times 10^D.$$

O enunciado garante que a resposta sempre é menor que 2^{31}, logo todo cálculo pode ser feito com variáveis do tipo `int` (inteiro de 32-bits).

Um caso que deveria ser tratado separadamente é o caso $C = D = 0$. Tal caso está dentro das restrições do enunciado e sua resposta é zero, pois uma placa não pode ser vazia.

Problema M: Removendo moedas no Kem Kradãn

Autor do problema: Marcos Kawakami
Análise: Marcos Kawakami

A condição de existência de solução para o jogo das moedas é surpreendentemente simples: Existe solução se e somente se o número de moedas com a face dourada inicialmente voltada para cima (chamaremos estas simplesmente de moedas douradas) for ímpar. Vamos dividir a prova deste fato em duas partes. Primeiramente provaremos que, se o número de moedas douradas for inicialmente ímpar, então existe solução. Depois, provaremos que não é possível remover todas as moedas se o número inicial de moedas douradas for par.

Provaremos a primeira parte por indução. Não é difícil perceber que, se o número de moedas douradas for inicialmente 1, é possível remover todas as moedas do jogo. Suponha agora que o número de moedas douradas seja $K > 1$ e ímpar. Considere a remoção da moeda dourada mais à esquerda. À esquerda desta moeda retirada, ou teremos um segmento contendo uma única moeda dourada, ou não teremos segmento algum. À direita desta moeda, dependendo da cor da face da moeda imediatamente à direita, podemos ter um segmento contendo K (se a face for branca) ou $K - 2$ (se a face for dourada) moedas douradas. Note que, caso o segmento tenha K moedas douradas, podemos repetir o passo para este segmento e, como o número de moedas brancas é limitado, eventualmente obteremos um segmento com $K - 2$ moedas douradas. Teremos, após este processo, um segmento com $K - 2$ moedas douradas e zero ou mais segmentos com uma única moeda dourada. Como cada segmento pode ser resolvido de forma independente, a configuração inicial admite solução.

A prova de que uma configuração com número par de moedas douradas não tem solução é bastante similar ao caso ímpar. Claramente um segmento sem moedas douradas não tem solução, pois nenhuma moeda pode ser retirada neste caso. Se o número de moedas douradas for par maior que zero, percebe-se que, independentemente da escolha da moeda dourada a ser retirada, sempre teremos como resultado ao menos um segmento não vazio também com número par de moedas. Dessa forma, qualquer sequencia de retiradas levará ao surgimento de um segmento sem moedas douradas, e portanto não há solução para este caso.

A primeira parte da prova nos dá um algoritmo linear para encontrar uma sequência válida de remoções: Retirar sempre a moeda dourada mais à esquerda. Mas é claro que esta solução não é sempre única. De fato, se enumerarmos somente as moedas douradas de 1 a K, podemos retirar qualquer moeda de índice ímpar pois os segmentos resultantes sempre terão um número ímpar de moedas douradas.