

## O caminho das pedras

Você está explorando um país distante e exótico, e ao chegar à beira de um rio cheio de jacarés você percebe que não há nenhuma ponte próxima. Felizmente você pode cruzá-lo pulando sobre um caminho de pedras que vão de uma margem à outra. Os nativos de um povoado próximo fizeram este caminho há muito tempo e colocaram as pedras a 1 metro umas das outras.

Infelizmente, com o passar dos anos (e a falta de investimentos em transporte e infra-estrutura em geral) algumas pedras afundaram na lama do rio ou foram afastadas de suas posições por animais. Nestes casos, você terá de pular por cima de um buraco, para uma pedra posterior. Felizmente, você está em forma e consegue dar pulos de 1, 2 ou 3 metros.

Infelizmente (de novo) os nativos cobram caro pra que se atravesse o caminho e você não tem a grana que eles pedem. Ao tentar negociar, eles oferecem:

— Diga quantos jeitos diferentes tem de dar os pulos e cruzar o rio e você pode passar de graça. Tentamos resolver este problema há anos e nunca conseguimos.

Sem muita alternativa, você topa o desafio e resolve contar todas as formas de cruzar o rio sem cair em uma posição vazia. Os dados de entrada são simples:

Uma string quase binária  $m101101001\dots010m$  com a letra  $m$  marcando uma margem e a outra margem, e entre as margens há  $n$  bits onde o bit  $b_i$  indica a existência ( $b_i = 1$ ) de uma pedra na posição  $i$  ou a sua falta ( $b_i = 0$ ). Então, você tem que encontrar o número de maneiras diferentes de pular de uma margem até a outra.

Nessa hora, os nativos completam:

— Ah sim. O pessoal da tribo anda meio gordinho e por isso depois de dar um pulo de 3 metros eles vão estar cansados, e o pulo seguinte não pode ser de 3 metros. Então não conte essas alternativas.

Você quer impressionar a tribo, afinal se fizer um serviço bem feito pode cruzar o rio e até ganhar uma arco e umas flechas! Para isso resolve fazer três versões de programa:

1. Uma versão recursiva bem simples para fornecer resultados iniciais;
2. Uma versão recursiva com memorização, pois você sabe que eles vão usar números grandes;
3. Uma versão não-recursiva, pois os computadores deles pode não ter a capacidade de fazer recursão.

**Faça seu programa receber a string quase binária pela linha de comando:**

```
> java saltos m10101100111m
Recursão simples:      existem 1234 maneiras
Recursão memorizada:  existem 1234 maneiras
Sem recursão:          existem 1234 maneiras

> java saltos m101111101010101011101111110111111010101111m
Recursão simples:      existem 123456 maneiras
Recursão memorizada:  existem 123456 maneiras
Sem recursão:          existem 123456 maneiras
```