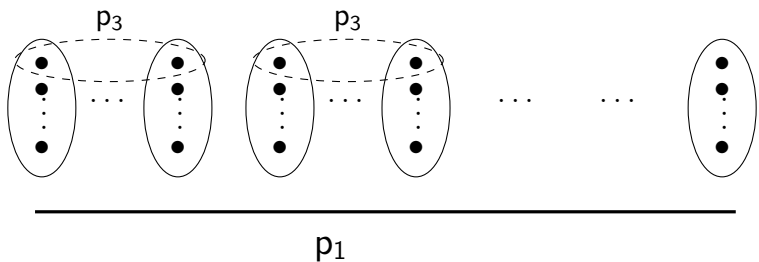


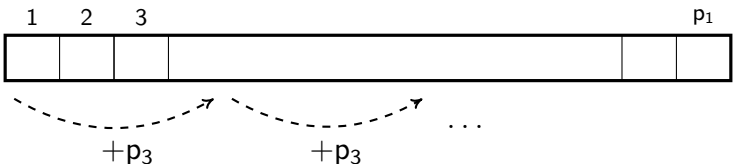
Outra propriedade legal dos primos

Certo.

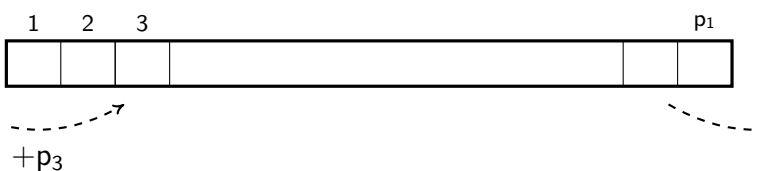
Mas a gente ainda pode tirar uma outra informação legal do nosso esquema



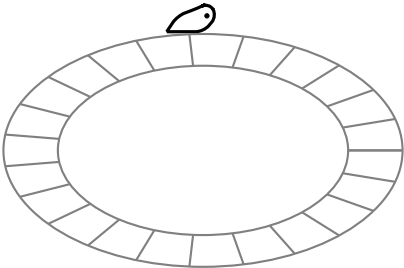
Quer dizer, formar grupinhos de tamanho p_3 nesse esquema é a mesma coisa que dar saltos de tamanho p_3 em uma lista de tamanho p_1



sendo que, na hora em que um salto ultrapassa o final da lista, a gente volta para o começo



E a gente também pode pensar em um sapo que dá voltas em torno de uma pista



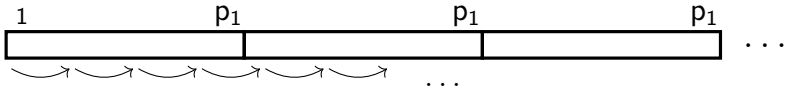
Daí, se a gente continua saltando e saltando dessa maneira, alguém poderia perguntar

- Quando é que, pela primeira vez, a gente cai em uma posição onde a gente já passou antes ?

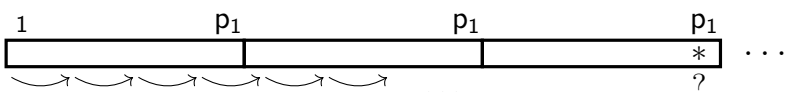
Bom, agora a gente precisa raciocinar.

E para fazer isso, nada melhor do que ver as coisas de um ângulo diferente.

Quer dizer, a gente pode imaginar que o processo salta sempre para frente, sobre uma sequência de cópias da lista



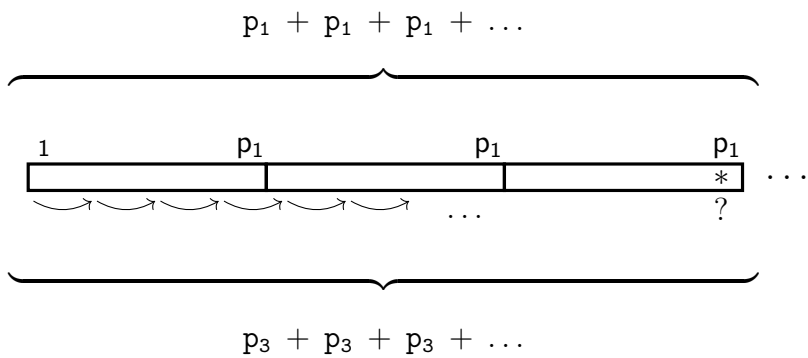
E daí, a gente pode perguntar: quando é que o processo cai na última posição (p_1) pela primeira vez ?



Mas isso a gente já sabe responder.

Quer dizer, quando o processo cai em p_1 , a distância total que ele percorreu até aquele momento corresponde a

- uma soma de saltos de tamanho p_3
- e também uma soma de listas de tamanho p_1



Em outras palavras, um múltiplo comum de p_1 e p_3 .

Ora, mas a gente já sabe que o menor múltiplo comum é $p_1 \cdot p_3$.

Daí que, a primeira visita à posição p_1 acontece após o processo ter percorrido a distância $p_1 \cdot p_3$

E isso corresponde a exatamente p_1 saltos.

Legal.

Agora a gente pode perguntar:

- Será que alguma posição foi visitada duas vezes antes disso ?

E a resposta é: claro que não!

Porque se o processo visitasse duas vezes a mesma posição ele entraria em repetição.

E daí, ele nunca iria visitar a posição p_1 .

Mas isso não pode ser.

Logo, nenhuma posição é visitada duas vezes nos primeiros p_1 saltos.

Ora, mas a lista possui p_1 posições.

Daí que, nos primeiros p_1 saltos o processo visita todas as posições da lista.

E logo em seguida, saltando a partir da posição p_1 ele cai na posição p_3

— (que foi onde tudo começou ...)

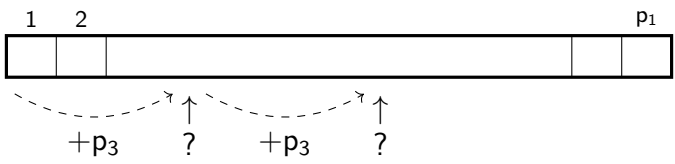
Não é legal?

E isso tem uma consequência ainda mais legal.

Quer dizer, imagine que você tem uma lista de tamanho p_1 .

E imagine que você quer saber se o elemento x está na lista ou não.

Então, ao invés de percorrer o elemento x percorrendo a lista elemento a elemento, a gente pode fazer isso saltando de p_3 em p_3 posições — (onde p_3 também é primo)



Daí, se você voltar ao lugar onde começou sem ter encontrado o elemento x , é porque ele não está lá.