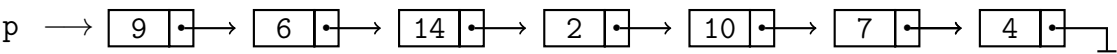


Imagine que o ponteiro **p** aponta para uma lista encadeada.

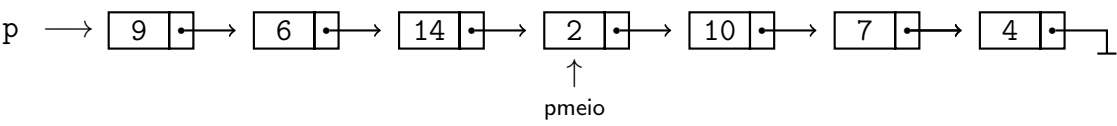
E imagine que a lista tem tamanho ímpar



A tarefa é

→ *Encontrar o elemento central da lista — (e apontar um dedo para ele)*

No exemplo acima, isso nos daria

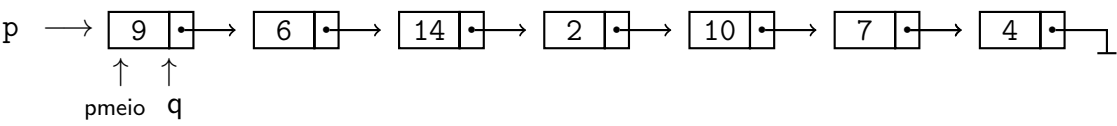


Bom, a estratégia padrão aqui é

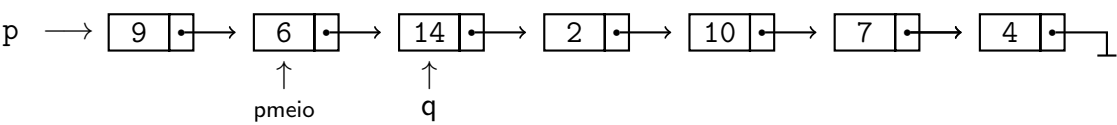
- percorrer a lista contando a quantidade de elementos
- calcular a posição do elemento central
- e depois levar o ponteiro **pmeio** até lá

Mas, a gente pode fazer uma coisa mais legal.

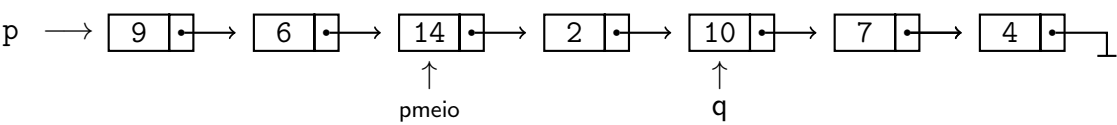
A ideia é colocar dois dedos no primeiro elemento



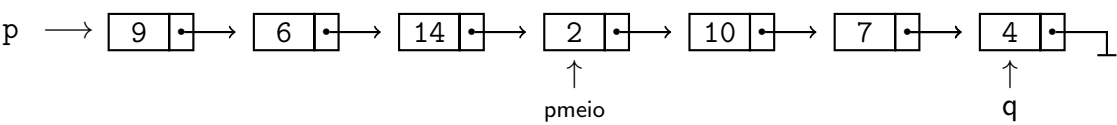
Daí, o ponteiro **q** dá um pulo de tamanho 2, enquanto o ponteiro **pmeio** dá um pulo de tamanho 1



A seguir, a gente faz a mesma coisa



E continua fazendo isso até que o ponteiro **q** alcance a última posição da lista



A coisa pára quando o ponteiro **q** não consegue mais dar o seu pulo de tamanho 2.

E nesse momento, o ponteiro **pmeio** está apontando para o elemento central.

Abaixo nós temos o código desse algoritmo

```
q = p;    pmeio = p                                // dois ponteiros no 1o elemento
Enquanto ( q != Nulo )
    q = q->prox                                     //
    Se ( q != Nulo )                               // pulo de tamanho 2
        q = q->prox                                 //
    Senão    Quebra                                // ou Quebra
    pmeio = pmeio->prox
Imprime('O elemento do meio é', pmeio->num)
```