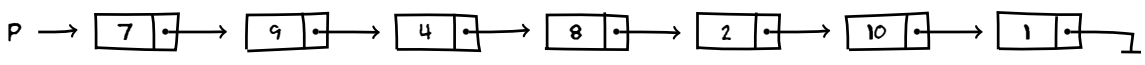
A operação de partição

Imagine que o ponteiro p aponta para uma lista encadeada



A tarefa é

→ *Colocar todos os elementos* $\leq$ `k` *no início da lista*
e todos os elementos $>$ `k` *no fim da lista*

No exemplo acima, para `k` = 7, isso poderia nos daria

Bom, isso é a mesma coisa que o exemplo anterior.

Quer dizer, basta considerar

- os elementos $\leq$ `k` como "elementos pares"
- os elementos $>$ `k` como "elementos ímpares"

Dessa vez, a gente vai fazer a coisa na volta — (*para preservar a ordem dos elementos*)

Então primeiro a gente vai até o fim da lista.

Daí, a chamada que vê o `Nulo` inicia o processo fazendo p1 = `Nulo` e p2 = `Nulo`.

E daí, cada chamada insere o seu elemento em p1 ou p2, dependendo do seu valor.

No final, basta concatenar as listas p_1 e p2.

A coisa fica assim,

```
func partição-Rec (q,k)

      Se (q == Nulo)   Retorna (Nulo,Nulo)

      p1,p2 = partição-Rec (q->prox,k)              // vai até o fim

      Se (q->num ≤ k)
            q->prox = p1;   p1 = q                  // insere na lista p1
      Senão
            q->prox = p2;   p2 = q                  // insere na lista p2

      Retorna (p1,p2)
```

Legal.

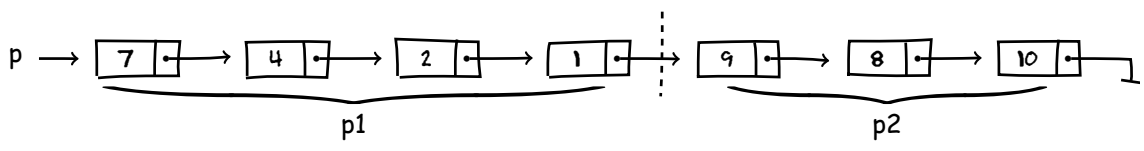
Mas, você viu que essa função não concatena p1 e p2?

Porque?

Bom, porque agora a gente vai fazer outra coisa.

O algoritmo Quicksort

A operação de partição (seguida da concatenação) já começa a colocar a lista em ordem



E para terminar a ordenação, basta continuar particionando p1 e p2 — (*é por isso que a gente não fez a concatenação*)

Mas aqui é preciso ter um pequeno cuidado.

Quer dizer, a partição foi feita com o número 7 — (*o primeiro elemento da lista*)

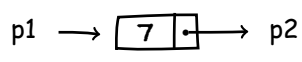
Isso significa que todos os outros elementos em p1 são menores que o 7.

E todos os elementos em p2 são maiores que o 7.

Logo, o lugar correto do 7 é entre as listas p1 e p2 — (*i.e.,* p1 *sem o primeiro elemento*)

Então, é conveniente remover o primeiro elemento da lista antes de fazer a partição — (*e utilizar o seu valor como pivô da partição*)

Daí, no momento da concatenação, a gente faz a coisa assim



— (*para obter a lista ordenada*)

Como todo mundo sabe, esse é o algoritmo Quicksort.

E a coisa fica assim

```
func quicksort-Rec (q)

      Se (q == Nulo)   Retorna (Nulo)
      Senão
            pri = q                                 // separa o 1o elemento
            p1,p2 = partição-Rec (q->prox,pri->num) // particiona
            p1 = quicksort-Rec (p1)                 // ordena os 2 lados
            p2 = quicksort-Rec (p2)                 //
            q = concatena (p1,pri,p2)               // e concatena
            Retorna (q)
```

◇