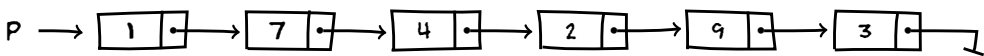


O último elemento no início

Imagine que o ponteiro p aponta para uma lista encadeada



A tarefa é

→ mover o último elemento para a primeira posição da lista

Bom, nós queremos fazer isso de maneira recursiva.

Então, a ideia é fazer as chamadas recursivas chegarem até o último elemento — (e parar ali)

Daí, a chamada que vê o último elemento retorna Nulo — (para a chamada anterior conectar o penúltimo elemento ao Nulo)

E também retornar um ponteiro para o elemento que era o último.

As outras chamadas apenas aplicam o truque da reconexão, e repassam o último elemento para trás.

Dessa maneira, o último elemento chega ao início da lista.

Mas, como é que uma chamada sabe que ela vê o primeiro elemento?

Bom, a função main() pode dizer isso para ela — (por meio de um parâmetro pri)

E daí, elas diz para as chamadas seguintes que elas não vêem o primeiro elemento.

A coisa fica assim

```
func UnI-Rec (q,pri) // Último no Início

    Se ( pri == 1 ) // a chamada que vê o 1o
        Se ( q == Nulo OU q->prox == Nulo)
            Retorna (q,Nulo) // já acabei ...
        Senão
            px,u = UnI-Rec (q->prox,0)
            q->prox = px // re-conexão
            u->prox = q
            Retorna (u,Nulo)

    Senão // demais chamadas
        Se (q->prox == Nulo) // último elemento
            Retorna (Nulo,q)
        Senão
            px,u = UnI-Rec (q->prox,0)
            q->prox = px // re-conexão
            Retorna (q,u)
```

E a função main() fica assim

```
func main ()

    p,u = UnI-Rec (p,1)
```