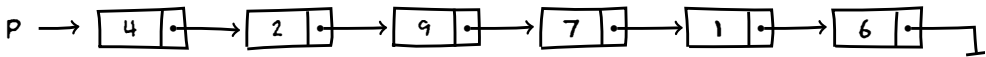


O maior elemento no fim

Imagine que o ponteiro p aponta para uma lista encadeada



A tarefa é

→ mover o maior elemento para a última posição da lista

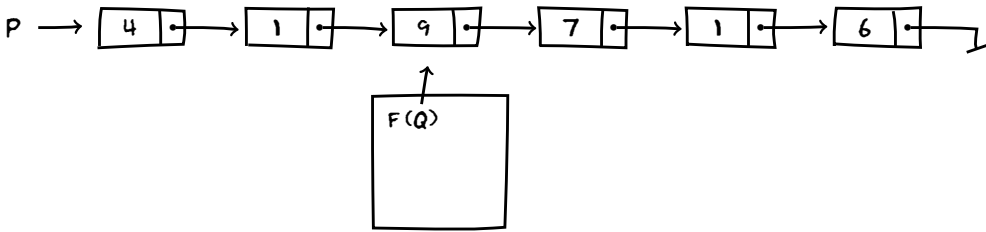
Bom, a ideia aqui é fazer a coisa em etapa outra vez

- 1. na ida a gente descobre que é o maior elemento
- 2. e na volta a gente encontra e manda ele para o fim

A primeira etapa vai ser feita da mesma maneira que no exemplo anterior.

Mas na segunda etapa a gente pode fazer uma esperteza.

Quer dizer, vamos ver as coisas na perspectiva da chamada que vê o maior elemento



O ponto é que essa chamada vê uma lista encadeada à sua direita — (apontada por q->prox)

E o que ela quer fazer é inserir um elemento no fim dessa lista — (o seu próprio elemento)

Ora mas se é assim, então tudo o que ela tem a fazer é chamar a função `inserçãoFim-Rec()`.

O único detalhe é que a função `inserçãoFim-Rec()` recebe um número como parâmetro.

Mas nós já temos um ponteiro para o registro que contém o número.

Então, nós vamos escrever uma outra função `inserçãoFimP-Rec()` para utilizar nesse exemplo.

A coisa fica assim

```
func MnF-Rec (q, Maa)                                     // Maior no Fim

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

    Senão
        Se ( q->num > Maa )    Maa = q->num
        px,M = MnF-Rec ( q->prox, Maa )
        Se ( q->num == M )
            q->prox = Nulo
            px = inserçãoFimP-Rec ( px, q )
            Retorna (px, -1)
        Senão
            q->prox = px
            Retorna (q, M)
```

A função `inserçãoFimP-Rec()` fica assim

```
func inserçãoFimP-Rec (q, r)

    Se ( q == Nulo )    Retorna (r)

    Senão
        px = inserçãoFim-Rec (q->prox, k)
        q->prox = px
        Retorna (q)
```

E a função `main()` fica assim

```
func main ()

    p,M = MnF-Rec (p, -1)
```