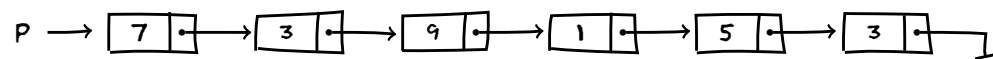


## O menor elemento no início

Imagine que o ponteiro `p` aponta para uma lista encadeada



A tarefa é

→ *mover o menor elemento para a primeira posição da lista*

Bom, aqui a gente precisa fazer a coisa em etapas

1. na ida a gente descobre quem é o menor elemento
2. e na volta a gente carrega ele para a primeira posição

Para descobrir que é o menor elemento, a gente utiliza um parâmetro `maa` que lembra quem é o menor elemento até aqui.

Daí, na chamada que vê o `Nulo`, esse parâmetro tem o valor do menor elemento.

Esse número é mandado de volta como valor de retorno.

Daí, a chamada que vê o menor elemento consegue saber disso — (*comparando o seu elemento com o valor de retorno*)

E então, ela remove o seu elemento da lista, e manda ele para o início da lista.

Finalmente, a chamada que vê o primeiro elemento faz a inserção do menor elemento no início da lista.

A coisa fica assim

```
func MnI-Rec (q,pri,maa)                                // Menor 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,m,pm = MnI-Rec (q->prox, 0, q->num)
            q->prox = px                                  // re-conexão
            pm->prox = q
            Retorna (pm, 0, Nulo)

    Senão                                                // demais chamadas
        Se (q->prox == Nulo)                             // último elemento
            Retorna (Nulo, maa, Nulo)
        Senão
            Se (q->num < maa)    maa = q->num              // atualiza maa
            px,m,pm = UnI-Rec (q->prox, 0)
            Se (q->num == maa)
                Retorna (px, -1, q)                      // retorna o menor elemento
            Senão
                q->prox = px                              // re-conexão
                Retorna (q, m, pm)
```

E a função `main()` fica assim

```
func main ()

    p,m,pm = MnI-Rec (p, 1, -1)
```