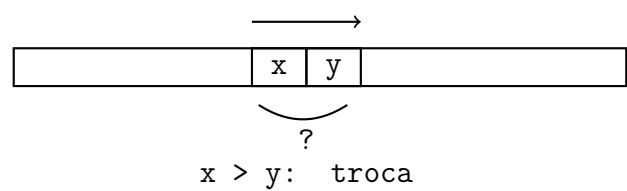
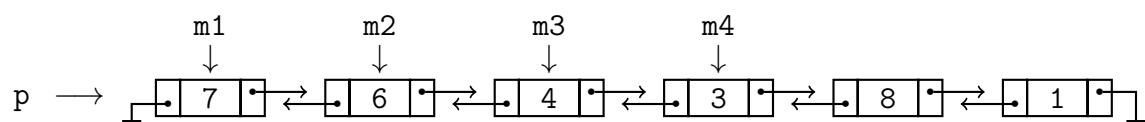


Lembre que a operação de varredura percorre a lista comparando elementos consecutivos, e fazendo a troca sempre que o elemento da esquerda é maior que o elemento da direita



Agora que temos a lista duplamente encadeada, nós podemos implementar essa operação de maneira mais natural — (*i.e.*, *percorrendo a lista e trocando os elementos de lugar*)

Para fazer isso, nós vamos utilizar 4 mãozinhas



Quer dizer, a cada passo nós comparamos os elementos apontados por `m2` e `m3`.
Daí, se o elemento apontado por `m2` for maior, as mãozinhas `m1` e `m4` nos ajudam a fazer a troca.
Depois disso, cada mãozinha dá um passo para frente.
A coisa fica assim

```
func Varredura (p)

    Se ( p == Nulo OU p->prox == Nulo )                // nada a fazer ...
        Retorna (p)

    m1 = Nulo;    m2 = p;    m3 = m2->prox;    m4 = m3->prox

    Enquanto ( m3 != Nulo )
        Se ( m2->num > m3->num )
            m2->ant = m3;    m3->prox = m2                // ajusta ponteiros de m2 e m3
            m2->prox = m4;    m3->ant = m1                //
            Se ( m1 == Nulo )    p = m3                    //
            Senão                m1->prox = m3            // ajusta ponteiros de m1 e m4
            Se ( m4 == Nulo )    m4->ant = m2                //
            aux = m2                                            //
            m2 = m3                                            // troca m2 e m3
            m3 = aux                                            //
            m1 = m2;    m2 = m3;    m3 = m4                // as mãozinhas andam
            Se ( m4 != Nulo )    m4 = m4->prox            //

    Retorna (p)
```