

Listas 2 - Ordenação

André Rauber Du Bois
dubois@inf.ufpel.edu.br
Computação - CDTec - UFPel

Exemplo: Insertion sort

- Para ordenar a lista

[7,3,9,2]

usando o **insertion sort**, retiramos o primeiro elemento e **ordenamos o tail** da lista obtendo

[2,3,9]

Para terminar a ordenação, precisamos inserir o **7** em sua posição correta na lista, obtendo

[2,3,7,9]

Insertion Sort

- Podemos definir esse algoritmo da seguinte forma:

```
defmodule InsertionSort do
  def insertion_sort([]) do
    []
  end

  def insertion_sort([h|t]) do
    ins(h,insertion_sort(t))
  end

  (...)
end
```

Insertion Sort

- Abordagem **top-down**: definimos o **insertion sort**, supondo a existência do **ins**
- Dividir o problema em partes menores, solucionando essas partes em separado, torna mais fácil a solução do problema como um todo
- Como ficaria a definição da função **ins**?
- **ins**: recebe um número e uma lista ordenada e insere o número na lista mantendo a mesma ordenada

```
defmodule InsertionSort do
```

```
  (...)
```

```
  def ins(x, []) do
```

```
    [x]
```

```
  end
```

```
  def ins(x, [h|t]) do
```

```
    cond do
```

```
      x <= h -> [x | [h|t]]
```

```
      true  -> [h|ins(x,t)]
```

```
    end
```

```
  end
```

```
end
```

```
iex(1)> InsertionSort.insertion_sort([7,3,2,9])  
[2, 3, 7, 9]
```

Exercícios

- Implemente a função **menor**, que usa o **insertion_sort** para encontrar o menor elemento de uma lista
- Implemente a função **maior**, que usa o **insertion_sort** para encontrar o maior valor em uma lista
- Modificando a implementação do **ins**, podemos modificar o comportamento da função **insertion_sort**. Implementar uma versão do **ins** que elimina elementos repetidos da lista. Implementar uma versão do **ins** que faz com que a lista seja ordenada de forma decrescente.