

## Tutorial: Inclusão de Subintervalos

Nesse problema, devemos ordenar os intervalos  $(a_i, b_i)$ :

- Em ordem crescente de  $a_i$ .
- Em caso de empate, em ordem decrescente de  $b_i$ .

Em seguida, devemos iterar sobre os intervalos ordenados, mantendo o maior valor de  $b_i$  visto até o momento. Se encontrarmos um intervalo  $(a_i, b_i)$  tal que  $b_i$  seja menor ou igual ao maior valor de  $b_i$  visto até o momento, então esse intervalo é coberto por um intervalo anterior e podemos descartá-lo. Caso contrário, atualizamos o maior valor de  $b_i$  visto até o momento.

O algoritmo abaixo implementa essa lógica.

`interval-cover`  
 $I$  Lista de intervalos  $I[0, n - 1]$  Lista de intervalos  $I'$  não cobertos por nenhum outro intervalo  
 $I' \leftarrow \emptyset$  `sort` $I$  Ordena os intervalos com critérios especificados  
 $\max_b \leftarrow -\infty$   $(a_i, b_i) \in I$   $b_i > \max_b$   
 $I'.append(a_i, b_i)$   $\max_b \leftarrow b_i$   $I'$

### Análise

O passo de ordenação tem complexidade  $\Theta(n \lg n)$  e domina o custo do algoritmo.