

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.

Algoritmo 1: INTERVAL-COVER(I)

Input: Lista de intervalos $I[0, n - 1]$

Output: Lista de intervalos I' não cobertos por nenhum outro intervalo

```
1  $I' \leftarrow \emptyset$ 
2 SORT( $I$ ) // Ordena os intervalos com critérios especificados
3  $\max_b \leftarrow -\infty$ 
4 foreach  $(a_i, b_i) \in I$  do
5   if  $(b_i > \max_b)$ 
6      $I'.\text{APPEND}((a_i, b_i))$ 
7      $\max_b \leftarrow b_i$ 
8 return  $I'$ 
```

Análise

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