

Tutorial: Inclusão de Subintervalos

Entendimento do Problema

O problema pede o menor número de intervalos necessários para que todos os outros estejam contidos em pelo menos um dos selecionados. Um intervalo $[l, r]$ cobre $[l', r']$ se, e somente se, $l \leq l'$ e $r \geq r'$. Isso significa que intervalos menores ou "internos" são redundantes se houver um intervalo maior que os englobe completamente.

Estratégia Ótima

A solução consiste em identificar os intervalos "maximais", ou seja, aqueles que não são cobertos por nenhum outro intervalo do conjunto. Se um intervalo não é coberto por ninguém, ele obrigatoriamente deve fazer parte de S' . Se ele é coberto por alguém, ele pode ser descartado.

Algoritmo Passo a Passo

1. **Ordenação Especial:** Ordene os intervalos S seguindo dois critérios:
 - Primeiro, pelo ponto de início l_i em ordem crescente.
 - Em caso de empate no início ($l_i = l_j$), ordene pelo ponto de fim r_i em ordem **decrecente**.
2. **Processamento Linear:** Após a ordenação, o primeiro intervalo da lista sempre será um candidato a S' (pois tem o menor l).
3. **Filtragem:** Mantenha uma variável **max_r** para rastrear o maior ponto de fim encontrado até agora entre os intervalos que decidimos manter.
 - Para cada intervalo $[l_i, r_i]$ na lista ordenada:
 - Se $r_i > \text{max_r}$, significa que este intervalo não é coberto por nenhum dos anteriores (pois, embora seu l_i seja maior ou igual, seu r_i estende a cobertura atual). Contabilizamos ele e atualizamos **max_r**.
 - Se $r_i \leq \text{max_r}$, este intervalo já está totalmente contido em algum intervalo anterior (que tinha $l \leq l_i$ e $r \geq r_i$). Portanto, ele é descartado.

Exemplo de Execução

Conjunto: $\{[2, 4], [3, 7], [1, 8], [4, 8]\}$

1. **Após Ordenar:** $[1, 8], [2, 4], [3, 7], [4, 8]$.
2. **Passo 1:** Pega $[1, 8]$. **max_r** = 8. (Total = 1).
3. **Passo 2:** $[2, 4]$. Como $4 \leq 8$, é redundante.
4. **Passo 3:** $[3, 7]$. Como $7 \leq 8$, é redundante.
5. **Passo 4:** $[4, 8]$. Como $8 \leq 8$, é redundante.
6. **Resultado:** 1.

Complexidade

- **Tempo:** $O(N \log N)$ devido à ordenação inicial. A varredura subsequente é $O(N)$.