Busca Binária - Estrutura da aula

Oficinas de Programação Competitiva

18 de novembro de 2025

Sumário

1 Explicação da busca binária - forma classica

https://codeforces.com/problemset/problem/2008/C (Longest Good Array) Aqui, o problema se resume a encontrar o maior x tal que $l + \frac{x*(x+1)}{2} < r$. Vamos argumentar que o melhor jeito de codar esse problema seria com busca binária.

Vamos começar com a analogia de encontrar uma palavra no dicionário. Vamos explicar como isso se aplica no problema que apresentamos e começar a falar de como codar. Nessa parte vamos definir como representamos o espaço de busca com l e r, e como atualizamos o espaço de busca a cada iteração, diminuindo o tamanho do espaço pela metade. Vamos argumentar que a complexidade disso será $O(\log n)$. Vamos mostrar o código e explicar quando podemos cair em um loop infinito, e como podemos arredondar pra cima para resolver. Aproveitando o gancho da codificação, que pode ser chata e tem vários casos de borda que temos que cuidar, vamos introduzir as funções da STL.

2 Explicação da busca binária - funções da STL

Aqui vamos começar mostrando a assinatura das funções lower bound e upper bound e explicar o que elas retornam. Em Seguida, vamos precisar explicar como usar os iteradores do C++ no contexto do lower bound e upper bound e os cuidados necessários para não cair em Runtime Error. Também vamos explicar como usar o lower bound ($\geq$) e upper bound ($>$) para obter os ($\leq$) e ($<$).

Em seguida, vamos apresentar um segundo problema: https://cses.fi/problemset/task/1073 (Towers), em que a função upper bound pode ser usada para facilitar bastante a codificação.

3 Explicação de busca binária na resposta

Aqui, vamos mostrar como a busca binária pode ser usada em funções, ao invés de vetores, desde que seja monotônica. Vamos apresentar o problema https://cses.fi/problemset/task/1620 (Factory Machines) como motivador. Aqui, vamos explicar como, usando essa técnica, conseguimos transformar problemas de otimização em problemas de decisão, que costumam ser mais

fáceis. Vamos mostrar o código e como podemos criar uma função e usá-la dentro da implementação clássica da busca binária.

4 Contest

Pensamos em termos 6 ou 7 problemas no contest. Gostaríamos que os 3 primeiros fossem os que apresentamos durante a aula para eles praticarem a codificação. Então, precisamos escolher outros 3 ou 4 problemas.

5 Kahoot

Podemos nos basear no quiz do USACO (https://usaco.guide/silver/binary-search)